NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
NWU-CS-04-32
May 7, 2004

Understanding Natural Language Descriptions of Physical Phenomena

Sven E. Kuehne

Abstract

The fact that human readers can learn about the physical world from textual descriptions leads to a number of interesting questions about the connections between our conceptual understanding of the physical world and how it is reflected in natural language. This thesis investigates some forms in which information about physical phenomena is typically expressed in natural language and how this knowledge can be used to construct models of the underlying physical processes.

Based on an analysis of the representations of physical quantities in natural language and common, reoccurring syntactic patterns, we implemented a system that uses Qualitative Process (QP) Theory to guide the semantic interpretation process to capture information about physical phenomena found in natural language text.

We have recast QP Theory in terms of frame semantics as FrameNet-compatible representations (QP frames) and use an extendable, controlled subset of English to capture QP specific information from natural language descriptions. In addition to general background knowledge based on a subset of the Cyc knowledge base and the lexical information supplied by a syntactic parser, the semantics of QP Theory are used in rules that guide the semantic interpretation process and the construction of QP Frames.

The thesis illustrates that QP Theory, as an established theoretical framework for handling continuous parameters and causation, can provide an essential component of natural language semantics.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
07 MAY 2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Understanding Natural Language Descriptions of Physical Phenomena	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Northwestern University,Computer Science Department,2133 Sheridan Road,Evanston,IL,60201**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**235**	

This research was supported by the Artifical Intelligence program of the Office of Naval Research.

Keywords:
Qualitative Process theory, natural language semantics, frame semantics, knowledge bases, Cyc.

This technial report is the reformatted version of:

Kuehne, S. E. (2004). *Understanding Natural Language Descriptions of Physical Phenomena.* Ph.D. thesis, Northwestern University, Evanston, IL.

NORTHWESTERN UNIVERSITY

Understanding Natural Language Descriptions of Physical Phenomena

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

by

Sven Erik Kuehne

EVANSTON, ILLINOIS

June 2004

ABSTRACT

Understanding Natural Language Descriptions of Physical Phenomena

Sven Erik Kuehne

The fact that human readers can learn about the physical world from textual descriptions leads to a number of interesting questions about the connections between our conceptual understanding of the physical world and how it is reflected in natural language. This thesis investigates some forms in which information about physical phenomena is typically expressed in natural language and how this knowledge can be used to construct models of the underlying physical processes.

Based on an analysis of the representations of physical quantities in natural language and common, reoccurring syntactic patterns, we implemented a system that uses Qualitative Process (QP) Theory to guide the semantic interpretation process to capture information about physical phenomena found in natural language text.

We have recast QP Theory in terms of frame semantics as FrameNet-compatible representations (QP frames) and use an extendable, controlled subset of English to capture QP specific information from natural language descriptions. In addition to general background knowledge based on a subset of the Cyc knowledge base and the lexical information supplied by a syntactic parser, the semantics of QP Theory are used in rules that guide the semantic interpretation process and the construction of QP Frames.

The thesis illustrates that QP Theory, as an established theoretical framework for handling continuous parameters and causation, can provide an essential component of natural language semantics.

Acknowledgements

Although there is only one name on the title page of this document, writing a dissertation is rarely a solitary enterprise. Over the past years, many people influenced the outcome of this work, directly or indirectly, sometimes even unwittingly, by providing ideas, critique, and inspiration. At this point, I would like to express my gratitude to a few people who have played an instrumental role in the completion of this journey that started more than seven years ago.

First of all, I would like to thank my advisor Ken Forbus for all the support he has given me, especially in times when others and even myself doubted the course and success of this project. He always left more than enough room to explore new ideas and provided guidance at exactly the right moments. I'd also like to thank Dedre Gentner, who gave me the unique opportunity to explore a number of interesting research areas that will be important for the future direction of this project. The members of her lab provided a great sounding board for new ideas, many of which have ultimately found their way into this thesis. Larry Birnbaum and Chris Riesbeck, my other two committee members, played key roles in the first wave of deep semantic natural language processing. They were an invaluable source of advice and always available for any questions.

Also, thanks to Chris Kennedy for his comments and suggestions on the parts of the thesis that address specific linguistic issues, to James Allen for making the source code his parser available to us, and to the folks at Cycorp for letting us use the contents of their knowledge base.

I definitely have to say thanks to Mike Brokowski, Ron Ferguson, Rob Harris, Dac Le, Joyce Ma, Tom Mostek, Praveen Paritosh, Jeff Usher, and Jin Yan, who have patiently endured and shared my moods, from euphoria to frustration, from bliss to bitterness, from care to cynicism, and back again. I could have graduated much sooner without all those discussions about nothing and everything, but it would have been quite boring.

And finally, I am most grateful to my family for all the support, love, and encouragement they gave me throughout the years I've spent here at Northwestern. This thesis is dedicated to Karl Alles, who first got me interested in science many years ago. Without him, I probably would have never taken this path.

Sven E. Kuehne
Evanston, May 2004

Table of Contents

Acknowledgements vi
Table of Contents vii
Table of Figures xi
Table of Tables xii
Chapter 1 Introduction 1
1.1 Capturing knowledge about physical phenomena 3
1.2 Controlling the input language 4
1.3 Extracting information about physical phenomena 5
1.4 A roadmap for the reader 7
Chapter 2 The Representation of Physical Quantities in Natural Language 9
2.1 Physical quantities 10
2.1.1 Constituents of Physical Quantities 10
2.2 Physical quantities in NL text 11
2.2.1 Explicitly referenced quantities 11
2.2.2 Implicitly referenced quantities 13
2.3 A closer look at adjectives and adverbs 15
2.3.1 Quantity-specific adjectives and adverbs 16
2.3.2 Quantity-neutral adjectives and adverbs 18
2.3.3 Transformation 18
2.4 Representation of values in physical quantities 19
2.4.1 Comparison 20
2.4.2 Symbolic labels 20
2.4.3 Concrete numeric values and units 21
2.5 Representations of changes in physical quantities 22
2.5.1 Verbs with direct references to a quantity change 23
2.5.2 Verbs with directional prepositional phrases 23
2.5.3 Verbs in combination with quantity-specific adverbs 23
2.5.4 Nouns with direct references to change 24
2.5.5 Nouns with directional prepositional phrases 24
2.6 Summary 24
Chapter 3 QP constituents in Natural Language 27
3.1 Patterns for constituents of physical processes 28
3.1.1 Process names 28
3.1.2 Sub-/Superclasses of processes 28
3.1.3 Participants 28
3.1.4 Conditions 29
3.1.5 Ordinal Relations 30
3.1.6 Miscellaneous Antecedent Relations 32

3.1.7 Direct Influences 32
3.1.8 Indirect Influences 34
3.1.9 Miscellaneous Consequence Relations 35
3.2 Landmarks and limit points 36
3.3 Summary 37
Chapter 4 QP Frames – a link between Natural Language and Qualitative Process Theory 38
4.1 Frame Semantics 38
4.2 QP Frames 41
4.2.1 The Quantity frame 42
4.2.2 The OrdinalRelation frame 42
4.2.3 The Influence frame 43
4.2.4 The QuantityTransfer Frame 44
4.2.5 Processes and their occurrences 45
4.3 Integration of QP Frames into the Cyc KB 46
4.4 Capturing NL information in QP frames 48
4.4.1 Example 1 48
4.4.2 Example 2 52
4.5 Summary 54
Chapter 5 QRG Controlled English – a controlled language for descriptions of physical phenomena 55
5.1 Types of controlled languages 56
5.2 The Parser 59
5.2.1 The Lexicon 61
5.3 Describing physical processes in natural language 66
5.3.1 Support of the constituents of QP Theory 66
5.3.2 General syntactic constructs 71
5.3.3 Limitations of QRG-CE 75
5.4 Examples 77
5.4.1 Fluid flow between containers 78
5.4.2 Conduction heat flow – the ice cube on a metal rod 78
5.5 Summary 79
Chapter 6 Semantic Interpretation 81
6.1 The parse-level semantic interpretation 82
6.1.1 Aligning the parser lexicon with the Cyc knowledge base 82
6.1.2 Retrieving semantic information for terminal nodes 83
6.1.3 Representations for choices between word senses 85
6.1.4 Combining semantic information in phrase nodes 86
6.1.5 Processing semantic information from parse trees 87

6.2 Evidence-based Word-Sense Disambiguation 88
6.2.1 Evidence-based word-sense disambiguation 89
6.2.2 Evaluation of evidence 94
6.2.3 Representations of resolved choice set information 94
6.3 Building QP frames 96
6.3.1 Quantity Frames 96
6.3.2 Changes in Quantities 98
6.3.3 Quantity Transfer frames 98
6.3.4 Direct Influence frames 100
6.3.5 Indirect Influence frames 100
6.3.6 Ordinal Relation frames 101
6.4 Merging QP frames 104
6.4.1 Sentence-level semantic interpretations 106
6.4.2 Paragraph-level semantic interpretations 109
6.5 Building process frames 110
6.5.1 Process frame rules 111
6.5.2 Constructing process frames 111
6.6 Summary 113
Chapter 7 Examples and Evaluation 114
7.1 Word-sense disambiguation and concept selection 114
7.2 Recognition of QP-specific information 118
7.2.1 Quantities 118
7.2.2 Indirect influences 121
7.2.3 Transfer between quantities 122
7.3 Merging frame information across sentences 124
7.4 Comparison against hand-coded models 127
7.4.1 Fluid flow between two containers 127
7.4.2 Conduction heat flow – ice cube, metal rod, and coffee 132
7.4.3 Other domains and types of processes 136
7.5 Rewriting and interpretation issues 137
7.6 Integration of linguistic and ontological resources 140
7.7 Summary 141
Chapter 8 Conclusions 143
8.1 Related work 144
8.1.1 Text understanding and Information extraction 145
8.1.2 Acquisition of lexical and conceptual knowledge 149
8.1.3 Ontologies and knowledge bases 150
8.1.4 Controlled languages and sublanguages 151
8.1.5 Parsing and Tagging 153
8.1.6 Semantic Interpretation 154

8.2 Future work 156
8.2.1 The background knowledge 157
8.2.2 The controlled language 157
8.2.3 The parser 158
8.2.4 The semantic interpreter 159
8.3 Outlook 160
References … 161
Appendix A Natural language patterns for QP constituents 181
A.1 Patterns for Indirect Influences 181
A.1.1 II1: THE x-er/THE y-er 181
A.1.2 II2: AS x, y 184
A.1.3 II3: WHEN x, y 187
A.1.4 II4: VERB PATTERNS 188
A.2 Patterns for Direct Influences 193
A.2.1 DI1: Transfer between quantities (active voice) 193
A.2.2 DI2: Transfer between quantities (passive voice) 197
A.2.3 DI3: Explicitly mentioned transfer event 199
A.2.4 DI4: Quantity change in object (active voice) 200
A.2.5 DI5: Quantity change in object (passive voice) 202
A.3 Patterns for Ordinal Relations 203
A.3.1 OR1, OR2: Difference comparison between quantities 203
A.3.2 OR3, OR4: Comparison between quantities 207
A.3.3 OR5: Adjective combination 211
A.4 Landmarks and limit points 213
A.4.1 L1: Action at a point 213
A.4.2 L2: Quantity at a point 214
A.4.3 L3: Conditional for points and intervals 215
A.4.4 L4: Labeling 215
Appendix B Rewrite Material 216
B.1 Example 1 216
B.2 Example 2 217
B.3 Example 3 218
B.4 Example 4 218
B.5 Example 5 219
B.6 Example 6 220
B.7 Example 7 220
B.8 Example 8 221
B.9 Example 9 222
B.10 Example 10 222

Table of Figures

Figure 1.1: Overview of the implemented system ……3
Figure 2.1: Dixon's Adjective Types ……15
Figure 3.1: Patterns for Ordinal Relations ……31
Figure 3.2: Patterns for Direct Influences ……33
Figure 3.3: Patterns for Indirect Influences ……35
Figure 3.4: Patterns for landmarks and limit points ……36
Figure 4.1: The FrameNet Motion frame ……39
Figure 4.2: The FrameNet Fluidic_Motion frame ……40
Figure 4.3: QP frames for a quantity transfer – Example 1 ……49
Figure 4.4: QP frames for an ordinal relation – Example 1 ……50
Figure 4.5: Process frame – Example 1 ……51
Figure 4.6: Process model – Example 1 ……51
Figure 4.7: QP frames for indirect influences – Example 2 ……52
Figure 4.8: Process frame – Example 2 ……53
Figure 4.9: Process model – Example 2 ……53
Figure 5.1: Layers of constraints in controlled languages ……59
Figure 5.2: Parse tree for 'The water is hot.' ……60
Figure 5.3: Lexicon entry format ……61
Figure 5.4: Grammar rules ……68
Figure 5.5: Grammatical constraints in QRG-CE ……71
Figure 6.1: Overview of the Semantic Interpretation Process ……81
Figure 6.2: Mappings between lexicon entries ……83
Figure 6.3: Retrieving semantic information for terminal nodes ……85
Figure 6.4: Intra-sentential merge algorithm ……106
Figure 6.5: Inter-sentential merge algorithm ……109
Figure 6.6: Interpretation Data ……112
Figure 7.1: Parse tree for 'The pressure in the cylinder is increasing' ……120
Figure 7.2: QP Frames for 'The Heat flows from the hot brick.' ……125
Figure 7.3: QP Frames for 'The heat flows to the cool ground.' ……126
Figure 7.4: QP Frames for merged interpretations ……127
Figure 7.5: QP Frames for two-container fluid flow ……128
Figure 7.6: Model fragment and scenario for two-container flow process ……131
Figure 7.7: QP Frames for conduction heat flow ……133
Figure 7.8: Model fragment and scenario for heat flow example ……136

Table of Tables

Table 7.1: Types of evidence and their weights ...……..115
Table 7.2: Coverage of entries ……..…….116
Table 7.3: Parts of speech for choice sets ……...……...…….116
Table 7.4: Number of choices per choice set ……...……...……117
Table 7.5: Evaluation of choice sets ……..……...…….117

Chapter 1

Introduction

Ordinary people know a lot about the physical world around them. They know that water will eventually boil if you heat it on a stove, that a ball placed at the top of a ramp will roll down, and that a cup will eventually overflow if you continue pouring coffee in it. People know all these things and can explain them with ease to others, but in most cases mathematical formulas are not a part of these explanations.

Instead of producing mathematical formulas or using formal representation languages, people use their own natural language to describe the physical world around them. Textbook writers introduce physical phenomena to students in plain English and use formulas after the important facts have already been stated in natural language. Authors of popular science books typically do not confront their readers with formulas at all. Depending on their target audience, they provide more or less detailed descriptions of the important facts and phenomena. The emphasis in all these cases is on developing a conceptual understanding of the phenomena.
The fact that human readers can learn about the physical world from textual descriptions leads to a number of interesting questions about the connections between the conceptual understanding of the physical world and how it is reflected in natural language. How is information about physical phenomena typically expressed in natural language? What are the connections between representations of physical processes and their corresponding realizations in natural language? If students can learn from simple descriptions of physical phenomena, can the knowledge included in these descriptions be extracted to automatically construct models of the underlying physical processes?

This thesis shows that Qualitative Process Theory (Forbus, 1984), as an established formalism for expressing mental models of physical phenomena, is an important component of natural language semantics. The claim is that understanding the connections between the ideas of Qualitative Process Theory and their manifestation in natural language descriptions provides insight in how knowledge about physical processes is communicated and how this knowledge can be captured in structured representations.

In particular, this thesis concerns the following three aspects of the relationship between QP Theory and natural language semantics:

1. *The constituents of Qualitative Process Theory are an important part of natural language semantics for understanding how knowledge about physical processes is communicated.*
   Natural language[1] reflects the ideas of Qualitative Process Theory and contains a number of syntactic patterns and semantic relations that can be used to construct representations of physical phenomena. We can identify typical natural language patterns for the QP constituents and map them to the constituents of QP Theory.

2. *The correspondences between natural language and QP Theory can be used in the creation of a controlled language to describe physical phenomena.*
   The syntactic patterns of QP constituents in natural language can be used to guide the construction of a controlled language for descriptions of physical phenomena. This language reduces the amount of ambiguous information found in unrestricted natural language and supports typical patterns for the constituents of QP Theory to aid the semantic interpretation process.

3. *The semantics of QP Theory can be used a natural language interpretation process to capture information about physical processes and construct models of physical processes.*
   The semantic interpretation process demonstrates how QP Theory can be used to capture the constituents of physical processes from controlled language descriptions as structured representations. In addition to general background knowledge provided by a knowledge base and the lexical information supplied by the parser, domain-independent properties of QP Theory are used to guide the semantic interpretation process. Information extracted from controlled language descriptions of physical phenomena can be used to construct representations of the underlying physical processes that are comparable to hand-coded models.

We have implemented a system that can extract QP-related information from simple natural language descriptions of physical processes, such as flow and motion events, and construct models of the underlying processes. Figure 1.1 shows an overview of the system, which includes a bottom-up parser for a controlled language, a word-sense disambiguation module, and a semantic interpreter.

---

[1] We limit our investigation to English. However, other languages certainly do contain similar correspondences between elements of QP Theory and natural language.

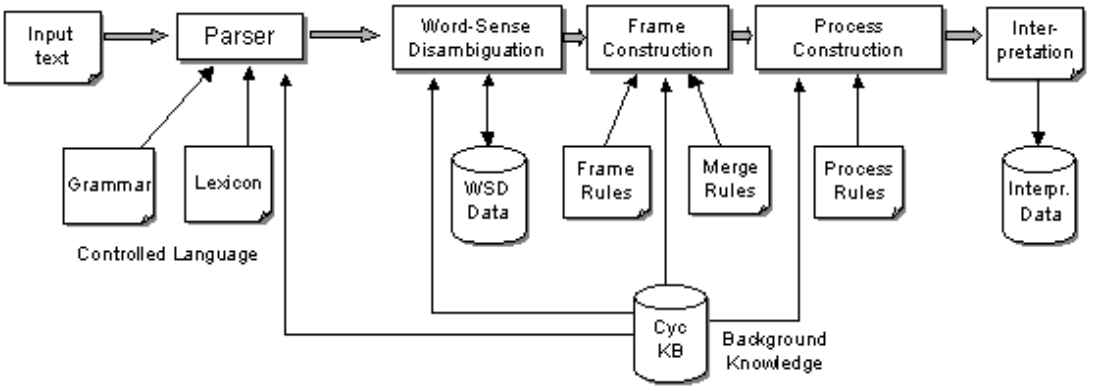

Figure 1.1: Overview of the implemented system

1.1 Capturing knowledge about physical phenomena

Textbooks, popular science books, and even dictionary-style summaries provide enough information for a conceptual understanding of many phenomena via examples. The skills of the intended audience, the purpose of the textbook, and assumptions about the background knowledge of its readers are some of the factors that determine the difficulty level of the text. A student reading a university textbook has very likely a different set of skills and different expectations than the reader of a popular science book.

The corpus material used for the research in this dissertation includes textbooks for middle school and university students as well as general popular science texts (Buckley, 1979; Maton et al., 1994; Moran & Morgan, 1994). With the exception of the university textbook, none of the sources contains equations describing physical processes. The writing style and the use of examples vary greatly between sources. Nevertheless, all these sources have two things in common: they use natural language to talk about the physical world, and they use examples to illustrate facts and formulas.

Understanding descriptions of physical phenomena starts with the identification of continuous parameters that are involved in the physical processes. Descriptions of physical phenomena typically contain abundant references to physical quantities. The extraction of information about continuous parameters is therefore an essential step in building models of physical processes. Chapter 2 of this thesis investigates the central role played by physical quantities.

If you look carefully at the descriptions of physical processes, given either as a concrete example or as generalized knowledge, you can find parts of the natural language description that correspond to certain elements of QP Theory. Sentences can contain a great amount of QP-related knowledge, as a corpus analysis (Kuehne & Forbus, 2002) has shown. The analysis of textbook descriptions of physical processes in terms of QP Theory was the starting point for the research described in this thesis and is reviewed in chapter 3.

We have recast QP Theory as a set of specialized frame structures (Kuehne & Forbus, 2002) that tie together the extracted words and phrases of the input text and their possible underlying QP semantics (Chapter 4). The frame structures are formally identical to QP Theory and allow the system to use standard QP reasoning techniques on these representations. Additionally, the semantics of Qualitative Process Theory provides constraints on the interpretation process.

QP Frames use a representational scheme that is compatible with the notions of frames and frame elements in FrameNet (Baker, Fillmore, & Lowe, 1998) (Fillmore, Wooters, & Baker, 2001). They form an intermediate representational layer between the natural language input and the representations that can be used in qualitative reasoning, e.g. model fragments in CML (Falkenhainer et al., 1994).

1.2 Controlling the input language

The fact that unrestricted natural language is full of ambiguity, even when the domain itself imposes some constraints, presents a challenge to any system that tries to extract information from text. A word can be *lexically ambiguous* by having multiple meanings and by being used in different parts of speech. In the following sentence the word ‘water’ is used in three different ways, i.e. as a noun, as a verb, and as part of a compound noun.

(1) The gardeners water the water lilies with water.

Syntactic ambiguity can arise from different prepositional phrase attachments, causing different interpretation of the sentence. A classic example is given sentence (2), which allows a number of interpretations, depending on who has the telescope and who is on the hill.

(2) I saw the man on the hill with the telescope.

A common approach to reduce ambiguity in natural language processing is to restrict the grammar and the lexicon by using a *controlled language*. We have designed QRG Controlled English (QRG-CE) as a controlled language for describing physical

phenomena in a readable, yet less ambiguous subset of English. It is based on an analysis of the syntactic realizations of components of physical processes in natural language and encodes these realizations as grammatical rules. The idea is to create a simple and easily usable controlled language for generating descriptions of physical phenomena.

Controlled languages have a long tradition that predates the fields of computational linguistics and natural language understanding (C. K. Ogden, 1933, 1937). More recently, controlled languages were used in diverse applications such as the generation of technical documentation for machinery (Almquist & Sagvall Hein, 1996) (Wojcik, Holmback, & Hoard, 1998), logic representations of operating procedures (Fuchs, Schwertel, & Schwitter, 1999) (Schwitter & Fuchs, 1996), and knowledge-based machine translation (Mitamura & Nyberg, 1995) (Nyberg & Mitamura, 1996).

Documents in unrestricted English need to be rewritten using the grammar and lexicon of the controlled language, but rewriting a document is easier (and the rewritten documents more readable) for a human author if the language is more habitable and allows a variety of syntactic realizations for the same underlying semantic construct. Consider the following two sentences (3) and (4), which are both supported by QRG-CE.

(3) The car is faster than the truck.
(4) The speed of the car is higher than the speed of the truck.

The semantic information extracted from sentences (3) and (4) should be identical. In both cases an ordinal relationship between the speed of the car and the truck should be constructed. Sentence (3) uses a more compact form and ‘hides’ the quantity type (speed, or velocity) and the ordinal relation in the comparative, while this information is made explicit in (4). Based on a corpus analysis of the forms in which QP-related knowledge can appear in natural language text (Chapters 2 and 3), we designed QRG-CE to allow a number of different syntactic forms in which semantically identical information can be expressed. However, some tradeoffs have to be considered with the use of a controlled language, since restrictions on the grammar and the lexicon limit the *expressiveness* and reduce the *habitability* of the language (Chapter 5).

1.3 Extracting information about physical phenomena

Since QRG-CE allows a certain degree of lexical and syntactic ambiguity to make the language more habitable, a semantic interpretation process is needed to eliminate any remaining ambiguity and produce the best possible interpretation of a sentence. This includes the disambiguation of multiple word meanings and the preference for domain-specific constructs. When students read descriptions of physical phenomena in

textbooks, they usually have certain expectations about the information they read and mentally construct appropriate models of these phenomena. This construction is an idiosyncratic process, because the students need to interpret the author's description and then build their own model of it. The student has to use available background knowledge to eliminate potentially ambiguous interpretations and to fill gaps left by the natural language description. In other words, reading about physical processes involves interpreting the text and constructing a model. In the best case, it is an exact reconstruction of the author's model of the process. The implemented system presented in this dissertation models this process by reading in textual descriptions of physical phenomena, parsing and interpreting the sentences, and reconstructing as best as it can the intended model in terms of a formal representation.

A *bottom-up parser* for descriptions of physical processes writing in QRG-CE produces a syntactic parse tree and a *general semantic interpretation* of the sentence (Chapter 6). This interpretation is constructed from semantic background information attached to the individual constituents of the sentence and combined according to the information given by syntactic parse tree. We make use of external resources, primarily the contents of the Cyc knowledge base (Lenat & Guha, 1989), to generate a general semantic interpretation.

Ambiguous conceptual information included in the general semantic interpretation data is resolved by a *word-sense disambiguation* module. For example, the semantic information attached to the noun 'bar' can include the concepts corresponding to 'drinking establishment' and 'unit of pressure'. Based on evidence such as contextual information and domain-specific constraints, the word-sense disambiguation process will prefer one concept over another. Using third-party provided resources such as the COMLEX lexicon and the contents of the Cyc knowledge base, we have to deal with inconsistencies such as missing entries, non-aligned argument structures and erroneous part of speech information. For this reason the word-sense disambiguation process uses an evidence-based approach that will collect and weigh various types of evidence supporting a word sense.

A *QP-specific semantic interpretation* step is used to construct QP frame structures from the general semantic information via sets of forward-chaining rules. This part of the semantic interpretation process also includes merging information from multiple sentences to generate a paragraph-level semantic interpretation. The semantic interpreter generates a set of QP frame structures as a representation of the underlying physical processes.

The output of the system can be evaluated by three different criteria: (1) *concept selection*, (2) *recognition of QP-specific information*, and (3) *coverage of automatically generated process frames in comparison to hand-coded models*.

Concept selection, i.e. the selection of the correct concepts for an individual word by the semantic interpretation process, allows predictions about the coverage of the background knowledge base and the ability of the word-sense disambiguation process. The recognition of QP-specific information can be shown by the ability of the controlled grammar and the semantic interpretation rules to identify QP-related information in the input text and to construct the appropriate representations. Finally, the QP-related content captured in the automatically generated representations can be evaluated by comparing the frame information constructed by the semantic interpreter against hand-generated models of physical processes.

1.4 A roadmap for the reader

The following chapter takes a closer look at the various ways in which information about physical quantities can appear in natural language text. Continuous parameters are a fundamental element of Qualitative Physics and provide basic information for the interpretation of descriptions of physical processes (Kuehne, 2003). Because of this central role, the identification of information about physical quantities is an essential task for a system that tries to build process models from natural language text.

Chapter 3 provides an in-depth analysis of the different forms in which constituents of Qualitative Physics are expressed in natural language. A corpus analysis previously reported in (Kuehne & Forbus, 2002) provided the starting point for this thesis. The chapter summarizes the results of this analysis and highlights the syntactic patterns in which constituents of Qualitative Physics can appear in natural language text.

In chapter 4 we introduce the representations used for storing information extracted from natural language text. *QP Frames* provide an intermediate representational layer between natural language and assertions for the background knowledge base. The use of these frame structures was motivated by the FrameNet project. QP frames are an extension of the ideas found in frame semantics.

Chapter 5 describes QRG Controlled English, the controlled language we designed for describing physical processes in natural language. The language provides support for the syntactic patterns identified in chapter 3 and aids the construction of QP frames during the semantic interpretation process.

With the identification of QP specific patterns, the definition of an intermediate representational layer and the introduction of the controlled language all the building blocks are in place to analyze and interpret natural language descriptions of physical processes. Chapter 6 describes the semantic interpretation process, which includes an evidence-based disambiguation of different word senses, the construction of QP

frames, an inter-sentential merge process to combine the information from several sentences for a paragraph-level interpretation, and the generation of process frames.

Illustrated by a number of examples, chapter 7 discusses the results of the semantic interpretation process in terms of three criteria: (1) the selection of the appropriate concepts from the background knowledge base, (2) the support and recognition of QP-specific information in the input text by the controlled language and the semantic interpretation rules, and (3) a comparison of automatically generated process frames against hand-coded expert models. Finally, chapter 8 takes a look at related research and lays out the plans for future extensions to our system.

Chapter 2

The Representation of Physical Quantities in Natural Language

When people talk and write about physical phenomena in everyday language, references to continuous properties are often part of their descriptions. From simple utterances like *"The coffee is hot."* to a more complicated comparison such as *"The average velocity of gas molecules is higher than the average velocity of molecules in a liquid."* being able to identify and extract the information about physical quantities is essential to understand these sentences. Using Qualitative Process Theory (Forbus, 1984) as the underlying formalism, this chapter investigates the forms in which information about continuous properties can appear in written natural language. The results of the analysis are used for the development of a controlled language in chapter 5 and in the semantic interpretation process in chapter 6.

Although our focus is on *physical* quantities found in descriptions of *physical* processes, such as expansion, movement, or transfer, the findings of this analysis are applicable to other types of quantities as well. Abstract and conceptual quantities are often referred to metaphorically by words with a physical basis and require a different

semantic interpretation. *"The price is hot."* does not have anything to do with temperature unlike *"The water is hot."* The techniques for the identification of information about such quantities are essentially the same.

The Qualitative Reasoning community has often assumed correspondences between the way in which information about continuous parameters and processes appears in natural language natural language and the representations of knowledge. These correspondences are indeed not accidental. Since Qualitative Process Theory is a formalism of how people reason about the physical world, the basic ideas of the Theory should be reflected in the language that people use to communicate their understanding of physical phenomena.

2.1 Physical quantities

In Qualitative Process Theory, all physical changes in continuous properties are caused by *physical processes*. The identification of continuous parameters is therefore an essential step in the extraction of information about physical processes from natural language text. In (Kuehne & Forbus, 2002) we have shown that natural language descriptions of physical processes can contain abundant information about the constituents of physical quantities, and we presented representational extensions to FrameNet (Baker et al., 1998; Fillmore et al., 2001) for capturing information about physical processes.

The examples presented in this chapter draw from the same corpus material used in the previous analysis (Buckley, 1979; Maton et al., 1994; Moran & Morgan, 1994). Some of the sentences have been shortened for clarity and simplified to highlight particular features.

2.1.1 Constituents of Physical Quantities

Information about continuous properties in natural language corresponds to the following five constituents of physical quantities:

- The *Entity* is a uniquely named object or an instance of a process associated with the quantity. For example, the word ‘brick’ in the noun phrase ‘the temperature of the brick’ denotes an entity. The noun ‘brick’ actually refers a particular individual, maybe ‘brick32’, not the collection of all bricks. Entities can also refer to difference parameters, e.g. the noun phrase ‘pressure difference’.
- The *Quantity Type* specifies the kind of parameter. The word ‘temperature’ in the noun phrase ‘the temperature of the brick’ is a reference to a quantity type.
- The *Value* specifies the numerical or symbolic value of the property. The number ‘3’ in the measure phrase ‘3 liters of water’ or the adjective ‘hot’ in the noun phrase ‘the hot ground’ are values associated with a quantity.
- The *Unit* specifies the physical units of the property. Example: The word ‘kilograms’ in ‘3 kilograms of lead.’ Units usually appear only in combination with a numerical value or with a quantifier.
- The *Sign of the Derivative* specifies how the parameter is changing. In the sentence “The temperature is increasing.” The sign of the derivative is expressed by the word ‘increasing’, which indicates that the parameter is changing in a positive direction.

Only the first two of these five constituents are required to identify a physical quantity. The quantity type together with the entity are sufficient to describe quantities like ‘the

temperature of a brick' or the 'the flow rate of heat'. Values, units, and information about changes are optional and often not explicitly stated.

Entities and quantity types can be named by unique labels. These labels are usually introduced together with the noun, e.g. 'the brick B1' or 'the pressure P32'. After it has been introduced, the label can than be used on its own, acting as a discourse variable that refers to the entity or quantity type. The patterns presented in this analysis do not use such labels, but are applicable to named entities and quantity types as well.

2.2 Physical quantities in NL text

Descriptions of physical phenomena often make abundant references to physical quantities (Kuehne & Forbus, 2002). This section shows some forms that are commonly used in natural language descriptions to express information about physical quantities. The analysis is mainly concerned about the different parts of speech in which information about physical quantities appears, not about the syntactic constructs represented by particular sentences.

2.2.1 Explicitly referenced quantities

Natural language text can refer to physical quantities either directly or indirectly, depending on whether the type of the quantity is explicitly mentioned in the sentence. *Explicit references* to quantities can be found in nouns, verbs, and adjectives that are morphologically related to quantity types.

2.2.1.1 Nouns

The quantity type can be explicitly mentioned as a noun, together with one or more entities that it is associated with.

(1) VOLUME flows from the *can* to the *ground*.
(2) The TEMPERATURE of the *brick* is rising.

Sentence 1 contains information about two physical quantities, the volume of some substance in the can and on the ground. The quantity type 'volume' is associated with both locations, i.e. the 'can' and the 'ground'.[1] In (2) the quantity type 'temperature' is associated with a single entity.

[1] In sentence 1, 'volume' stands in for the actual substance that flows from the can to the ground. The motivation of the author (Buckley, 1979) was probably to describe the transfer of volume in a similar way to a transfer of heat energy as in 'Heat flows from the stove to the kettle.'

The quantity type can also appear as the head of a compound noun. The remaining constituents of the compound noun can be treated as information about a specialization of the quantity type. For example, in (3) the quantity type ‘radiation heat’ is a specialization of ‘heat’; in (4) ‘heat energy’ is a type of ‘energy’.

(3) RADIATION HEAT flows from the *heater* to the *hand*.
(4) The HEAT ENERGY of the *water* increases.

2.2.1.2 Verbs

Verbs can refer to events as well as to quantity types associated with these events.[2] The verb in (5) appears as a direct reference to the quantity type ‘length’. Sentence (6) is slightly more complicated, because it allows two different interpretations. The obvious interpretation is to treat the verb as an explicit reference to a quantity, as it is in (5). In this case, the quantity type ‘heat’ is tied to both entities, the stove as the source of the heat flow and the kettle as the destination of the heat flow.

(5) The press LENGTHENS the *iron beam*.
(6) The stove HEATS the *kettle*.

Alternatively, the sentence could be interpreted as an increase in temperature of the kettle caused by the stove. Even though the quantity type ‘temperature’ is not mentioned in the sentence, we might infer that heating the kettle also increases the temperature of the kettle. This is an inference that most readers of such a descriptions draw, and it coincides with the kind of conclusions that are supported by QP Theory.

2.2.1.3 Adjectives

Certain adjectives can refer to quantity types directly, if the adjective is morphologically related to a quantity type. For example, in (7) the adjective ‘denser’ refers to the quantity type ‘density’. The quantity type in this sentence is associated with both entities, the subject ‘iron’ and the object ‘wood’. The quantity type referenced in (8) by the adjective ‘deep’ is ‘depth’ and associated with the noun ‘pit’.

(7) *Iron* is DENSER than *wood*.
(8) The DEEP *pit* is covered with dirt.

---

[2] Events such the increase or decrease of a parameter, e.g. the temperature of a brick, can be involved in an instance of a physical process. For an interesting linguistic perspective on actions, processes, and events, see (Parsons, 1990).

2.2.2 Implicitly referenced quantities

While the quantity types in explicitly referenced quantities are usually easy to determine, *implicit references* to quantities are more difficult to figure out. Implicitly referenced quantities do not mention a quantity type. Instead, the reader has to use the contextual information provided by the sentence as well as available background knowledge. The following section shows how nouns, verbs, adjectives, and adverbs can determine a quantity that is not explicitly mentioned in a sentence.

2.2.2.1 Verbs

A quantity type can be implicitly referenced by a verb that describes a physical process, e.g. movement, expansion, or transfer. The sentence in which the verb occurs usually provides additional contextual information for the interpretation of implicitly referenced quantities.

(9) As the temperature rises, the *liquid* EXPANDS.

The verb ‘expand’ in (9) indicates that something is changing in one or more physical dimensions, i.e. in length, area, or volume. For the three-dimensional entity ‘liquid’ the appropriate quantity type is therefore ‘volume’. The verb also includes implicit information about a positive change in the quantity, i.e. an increase in volume of the liquid, which we will address shortly.

2.2.2.2 Adjectives

The quantity type can be implicitly referenced by certain adjectives. For example, the quantity type described by the adjective ‘hot’ in (10) is ‘temperature’. The comparative also encodes the ordinal relationship between the quantities associated with the two entities, i.e. the fact that the temperature of the stone is greater than the temperature of the water. Similarly, the quantity type expressed by ‘lighter’ in (11) is ‘weight’.

(10) The *stone* is HOTTER than the *water*.
(11) The *upper air masses* are LIGHTER than the *lower air masses*.

For a correct interpretation the relationship between the adjective and the associated quantity type has to be known. The fact that the adjective ‘hot’ is associated with ‘temperature’ is a fact learned by a human reader. This information has to be provided

as background knowledge in an NLP system, either entered manually, explained to the system, or learned automatically.

2.2.2.3 Verb/Adverb combination

Quantity types can also be determined by combining verbs and adverbs. The quantity type referenced in (12) is the rate of movement, or 'velocity'. The adverb alone is not sufficient to determine the quantity type. Although 'faster' is generally associated with velocity, it just qualifies the rate of change, i.e. that something is happening in less time. There are cases in which the quantity type referenced by 'faster' is not velocity. For example, 'expanding faster' in (13) refers to the rate of expansion.

(12) The *gas molecules* are MOVING FASTER than *molecules in a solid*.
(13) *Liquid A* is EXPANDING FASTER than *liquid B.*

All these cases have one thing in common: the referenced quantity is a rate, most likely associated with a process referenced by the verb ('movement', 'expansion', 'decay').

2.2.2.4 Noun/Verb combination

Noun/verb combinations can implicitly refer to the rate of change of a quantity. The quantity type in (14) is not 'heat' but the flowrate of heat. The combination of 'flows' and 'heat' determines the quantity type, while the combination of 'flows' and 'harder' gives the direction of change.

(14) [The greater the thermal resistance,] the HARDER the *heat* FLOWS.
(15) [The less heat is supplied,] the SLOWER the *temperature* RISES.

Sentence (15) looks similar to (14) but differs in an important domain-specific way: temperature is not an extensive property, i.e. temperature cannot be added directly to an object. The quantity type referenced in (15) is the rate of change in temperature, resulting from a change in the amount of heat.

2.2.2.5 Noun/Adjective combination

The quantity type is only implicitly referenced by a combination of a noun and an adjective.

(16) The BIGGER the *surface* [is], [the more heat is absorbed.]

The quantity type in (16) is the size of the surface (not the surface itself) associated with an unnamed participant or the size of a participant ‘surface’. The adjective ‘bigger’ refers to the quantity type ‘size’ (or ‘area’). It also encodes a change of the quantity, i.e. an increase in surface area.

As in Verb/Adverb combinations, the adjective determines the referenced quantity type. For example, replacing ‘bigger’ with ‘shinier’ will change the resulting quantity type from ‘area’ to ‘reflectance’. The following section investigates the roles of adjectives and adverbs in determining implicitly referenced quantities in more detail.

2.3 A closer look at adjectives and adverbs

Adjectives and adverbs play a special role in the interpretation of quantity types. A change of the adverb in a Verb/Adverb combination or the adjective in a Noun/Adjective combination can completely change the interpretation of the underlying quantity type. Comparisons between two quantities can be presented by explicitly mentioning the quantity type or by using an adjective or adverb as an indirect reference to the quantity type. An important distinction can be made about how adjectives and adverbs encode information about references to physical quantities, i.e. whether they are tied to a *specific* type of quantity or are *neutral* in regard of a quantity reference.

Other research on the lexical semantics of adjectives has tried to establish taxonomies

1. **Dimension:** *big, great, thin, narrow*
2. **Physical Property:** *hard, strong, clean*
3. **Speed:** *quick, fast, rapid*
4. **Age:** *new, old, young*
5. **Color:** *white, black, red*
6. **Value:** *good, bad, odd, strange*
7. **Difficulty:** *easy, difficult, tough*
8. **Qualification**, with six subtypes:
   definite, possible, usual likely, sure, correct
9. **Human Propensity**, with six subtypes:
   fond, angry, happy, unsure, eager, clever
10. **Similarity**: *like, unlike, similar, different*

Figure 2.1: Dixon's Adjective Types

for the different semantic categories of adjectives (see Raskin & Nirenburg (1995) for an overview). Several of these taxonomies focus on the class of adjectives that we are most interested in for extracting information about physical quantities, i.e. qualitative (scalar, gradable) adjectives (Dixon, 1991; Frawley, 1992). However, none of these taxonomies has been used in any practical application until now (Raskin & Nirenburg, 1995).

Dixon's list of semantic types for adjectives appears to be one of the most representative taxonomies. His list consists of ten types (Figure 2.1), which differ in their grammatical properties. Frawley's taxonomy is a slight variation of Dixon's model and uses only six types. It has strong focus on 'Quantity' (with a in-depth description of quantifiers) and 'Physical Properties', which is subdivided into 'Sense', 'Consistency', 'Texture', 'Temperature', 'Edibility', 'Substantiality', and 'Configuration'.

From our perspective, using the semantics of Qualitative Process Theory, the taxonomies suggested by Dixon and Frawley are flawed and inconsistent. The breakup of types and subtypes appears to be arbitrary, because several of the types of quantities can be collapsed into a single type. In Dixon's taxonomy the adjectives of the 'Speed' and 'Physical Property' types are separated from those classified as 'Dimension'. Similarly, 'Age' and 'Value' are listed as separate types, while they could actually be treated as a single kind of quantity. Furthermore, labeling one of types a 'Dimension' and another 'Physical Property' is misleading. All dimensions are quantity types, but not all quantity types are dimensions.

Dixon and Frawley mention that their adjective types have different grammatical properties and show different syntactic behavior, which would suggest that these types are based on syntactic properties rather than being a semantic classification. We are not trying to redefine the syntactic classifications. Instead, our approach is driven by the semantics of QP Theory. We are dividing the class of qualitative (gradable, scalar) adjectives into the two distinct classes mentioned above: quantity-specific and quantity-neutral adjectives.

2.3.1 Quantity-specific adjectives and adverbs

Quantity-specific adjectives and adverbs encode information about an implicitly referenced quantity, i.e. the adjective or adverb determines (sometimes in combination with a noun or verb) the quantity type. Sentences that use quantity-specific adjectives and adverb do not contain explicitly referenced quantities. The information about the quantity type is encoded in the adjective or adverb itself and needs to be retrieved from there.

(17) The stone is HOTTER than the water.

The comparative ‘hotter’ in (17) refers to the quantity type ‘temperature’ that is associated with the two entities ‘stone’ and ‘water’. The reader has to know that the adjective ‘hot’ is associated with ‘temperature’ or otherwise the interpretation of the sentence would fail. The use of the comparative also imposes an ordering on the two physical quantities, i.e. the temperature of the stone is higher than the temperature of the water. Furthermore, the adjectives used in these types of comparison have to be gradable. Explicit quantity references in comparisons are usually not combined with quantity-specific adjectives, as it is illustrated by (18).

(18)* The *temperature* of the stone is HOTTER than the *temperature* of the water.

The additional information about the quantity type originating from the explicit reference would be considered redundant, when the quantity-specific adjective and the noun refer to the same quantity type.[3] In other words, the use of explicitly referenced quantities and quantity-specific adjectives and adverbs (for the same entity and referring to the same quantity type) should be mutually exclusive.

An important aspect of the use of quantity-specific adjectives and adverbs is their interaction with nouns and verbs, and the quantity type that is referenced by the combination of them. Using different quantity-specific adjectives and adverbs with the same noun or verb changes the implicitly referenced quantity type.[4]

(19) Gas molecules are MOVING FASTER than molecules of a liquid.

In (19), the adverb ‘faster’ together with the main verb determines the type of the indirectly referenced quantity. The adverb ‘fast’ is quantity-specific, while the main verb of the sentence is not. The combination of ‘to move’ and ‘fast’ refers to the quantity type ‘velocity’. However, replacing the quantity-specific adverb with another adverb of the same category will result in references to different quantity type, as demonstrated in the following variations of (19).

(20) *Gas molecules* are MOVING HIGHER than *molecules of a liquid.*
(21) *Gas molecules* are MOVING FARTHER APART than *molecules of a liquid*.

---

[3] If they refer to different quantity types, the sentence would be considered quite problematic, e.g. if we try to replace ‘temperature’ in sentence 18 with ‘weight’.
[4] It is assumed that the verb or noun in the combination does not include an explicit quantity reference, i.e. it has to be quantity-neutral.

The indirectly referenced quantities in (20) are the ‘height’ (or position) of the gas and liquid molecules, and the quantity type referred to in (21) is the ‘distance’ between the molecules.

2.3.2 Quantity-neutral adjectives and adverbs

Comparisons between two quantities do not always have to use quantity-specific adjectives and adverbs in their comparative form. Another class of adjectives and adverbs does not carry any quantity-determining information and is therefore labeled as *quantity-neutral*. Sentences with adverbs and adjectives of this class need to reference the quantity directly, because the quantity-neutral adjective or adverb does not contribute any information to determine the quantity type. In (22) the quantity type (‘temperature’) is explicitly mentioned for both entities (the ‘food’ and the ‘plate’). The comparison is done by a quantity-neutral adjective ‘high’.[5]

(22) The temperature of the food is HIGHER than the temperature of the plate.

The direct reference to the quantity type used in the comparison does not need to be included in a noun phrase with the entities. A common form of comparison references the quantity type as a part of a noun phrase that includes a quantity-neutral adjective.

(23) The *tub* has a GREATER *volume* than the *can*.

The explicitly referenced quantities in (23) are the ‘volume’ of the ‘tub’ and the ‘can’. The direct reference to ‘volume’ applies to both entities in this pattern. The quantity-neutral comparison does not contribute any information to determine the quantity type; it just determines the ordering between the two quantities.

2.3.3 Transformation

The sentences in the previous two sections referred to physical quantities by using either a combination of explicit references to quantities and quantity-neutral adjectives and adverbs, or a combination of implicit quantity references and quantity-specific adjectives and adverbs. These combinations should be mutually exclusive for the same quantity type and the same associated entities. The following example illustrates how sentences containing quantity-specific adjectives and adverbs (24, 25) can be rephrased in quantity-neutral forms (26, 27).

---

[5] The adjective ‘high’ can be used either in a quantity-specific sense (referring to ‘height’ or ‘depth’ as a quantity type), or in a quantity-neutral way (in the sense of ‘more’ or ‘greater’).

(24) The *stone* is HEAVIER than the *brick*.
(25) The *food* is HOTTER than the *plate*.
(26) The *temperature of the stone* is GREATER than the *temperature of the water*.
(27) The *stone* has a GREATER *weight* than the *brick*.

These examples suggest that sentences with quantity-specific adjectives and adverbs can be changed into their quantity-neutral counterparts, and that sentences with quantity-neutral adjectives and adverbs with explicit references to quantities in nouns have an alternative quantity-specific form. In other words, sentences using quantity-specific constructs can be transformed into semantically equivalent quantity-neutral constructs, and vice versa.

Being able to transform or rewrite sentences with implicit references to a quantity type into an equivalent form that makes the quantity type explicit and uses only generic, quantity-neutral adjectives and adverbs, is an important step towards the creation of a simplified grammar and the semantic interpretation of physical quantities.

2.4 Representation of values in physical quantities

The previous sections were primarily concerned about the information about the entity and quantity type, the two mandatory constituents of physical quantities. Knowing the type of a quantity and the entity it is associated with enables us to talk and reason about it. A simple noun phrase such as ‘the depth of the water’ contains enough information to recognize it as the description of a physical quantity, even without having any information about a particular value the quantity might have, the unit of that value, or the direction in which the quantity is changing. On the other hand, sentences like (28) could provide information for all five constituents.

(28) The temperature of the oil is rising to 250 degrees Fahrenheit.

The identification of the quantity type and the entity is just half the story when we are dealing with representations of physical quantities. The following two sections examine how values and units of quantities appear in natural language text, and how changes in quantities can be identified.

There are three common types of references to values and units that can be found in natural language text: (1) in the context of comparisons, (2) as symbolic labels, and (3)

as quantitative information. We will discuss values and units together because units usually appear in combination with values.[6]

2.4.1 Comparison

Values in the context of a comparison appear in sentences like "The brick is warmer than the plate." The comparison orders the values of the quantities, i.e. the temperature of the brick is greater than the temperature of the plate. However, it does not contain exact information about the possible values of the quantities. Even though the base form ('warm') of the comparative might refer to a specific range of temperature, the exact values cannot be known or even guessed from the information provided by the sentence. The brick might be red hot, while the plate is frosted with ice. This fact becomes more explicit if the quantity-neutral form of the sentence is used, "The temperature of the brick is higher than the temperature of the plate." Replacing the comparative 'warmer' with 'hotter' will not change the ordering between the quantities or contribute any additional information for identifying an exact value. However, the use of a weak comparative such as 'warmer' would violate felicity conditions if the difference between the two temperatures is extreme, and vice versa.

It is impossible to determine how far the values associated with the two compared quantities are apart from each other. The only information that can be extracted from this sentence about the values of the two compared quantities is the fact that the value of one quantity is greater than the other. With several of these comparisons along the same dimension, it is possible to identify the potential ranges of the values for particular quantities. For example, the temperature of coffee is greater than the temperature of an ice cube, but it is lower than the temperature at the tip of a lit cigarette.

2.4.2 Symbolic labels

Values can also take the form of a *symbolic label* associated with an entity, e.g. "The brick is hot." Even though the exact temperature of the brick is unknown, the adjective 'hot' suggests a certain temperature range. The range might be different depending on the context of the sentence. In refrigeration 'hot' might be in a very different range of temperatures than in the context of metallurgy.

Nouns that are associated with the adjective can impose restrictions on the range of the value in certain cases. For example, (Bierwisch, 1967) compares two simple

---

[6] Units can appear separately from values in definitional statements, like "Length is measured in Meters." or, even more explicitly, "The unit of power is the Watt."

sentences, "*The room is tall.*" and "*The space is tall.*" In the first sentence the noun 'room' might restrict the range of values for the height of a room to those for a typical room, e.g. between 8 and 10 feet. Without further information, this kind of assumption is more difficult to make for second sentence. Is the space a small compartment or a crawl space? Or is it the inside of a cathedral? The range of typical values would be quite different for these two cases.

The concepts of quantity-specific and quantity-neutral forms are applicable to these symbolic labels for values. Adjectives that represent a value are generally quantity-specific, as in the sentence "The brick is hot." Alternatively, a quantity-neutral form could be used to express the same fact, e.g. "The temperature of the brick is high."

While adjectives and adverbs generally refer to a range of values along a dimension, natural language also uses symbolic labels to refer to concrete values, i.e. particular points along a dimension. The noun phrase 'boiling point of water' usually refers to the point where liquid water turns into steam and the value of approximately 212 degrees Fahrenheit. The noun phrase provides a label for this particular point. Note that the compound noun 'boiling point' would be an underspecified symbolic label because different substances have different boiling points. Other labels such as 'sound barrier' may not need the additional complement.

The structure for labels that describe limit points is not arbitrary. Usually the head of a noun phrase refers to a point on a scale (e.g. 'point', 'barrier', 'threshold'), while the noun modifier is associated with a process, a dimension, or a quantity type (i.e. 'boiling', 'sound'). These two parts are mandatory components of the label. Determining the quantity type and the dimension is difficult in many cases, e.g. we have to know that 'boiling point' is associated with 'temperature' and that 'sound barrier' actually refers to the speed of sound or velocity. Additionally, the label can take an optional complement phrase that restricts the compound noun. For example, the complement phrase 'of water' restricts the interpretation of boiling point to a particular substance. The key idea here is that the underlying linguistic mechanisms for handling limit points are essentially the same as those for symbolic references to intervals on a particular dimension.

2.4.3 Concrete numeric values and units

The most explicit form in which values can appear is as *quantitative information*, i.e. by using concrete numeric values and units. For example, in (29) the quantity type ('temperature') is explicitly stated, together with exact information about the numeric value ('120') and the unit ('degrees Fahrenheit').

(29) The temperature of the brick is 120 degrees Fahrenheit.

Sentences that contain concrete numeric values and units usually do not use quantity-specific adjectives or adverbs in addition to a numeric value.

(30)* The water is 80 degrees Celsius hot.
(31) The water has 80 degrees Celsius.

Sentence 30 should be considered anomalous, because the adjective ‘hot’ provides at best redundant information in the form of a symbolic value. Units can refer indirectly to the quantity type that they are associated with, as in (31). The association between units and quantity types is a learned fact and has to be encoded as background knowledge in an NLP system.

2.5 Representations of changes in physical quantities

The values of physical quantities cannot always be treated as static information; they will change as physical processes are active. The sign of the derivative indicates whether a quantity is changing and in which direction. This section takes a look at how changes in physical quantities are reflected in natural language.

The most obvious choice to express changes in the physical world is the use of verbs. For example, if water is flowing from one container into another, there are several ways of expressing the change of the amount of water in each container. It could be explicitly stated that the amount of water in one container is decreasing while the amount of water in the other is increasing. Alternatively, one could say that water is flowing from one container to another, without ever mentioning the increase and decrease in the two involved quantities.

(32) The amount of water in container A is decreasing, while the amount of water in container B is increasing.
(33) Water flows from container A to container B.

Although these two sentences might be applicable to the same scenario, they are not equivalent. For example, (33) only implies a decrease of the amount of water in A. It does not state this information explicitly. On the other hand, (32) implies a flow, without actually mentioning it. These distinctions are important for a semantic interpretation process, because the information that is directly available from the sentences is different.

2.5.1 Verbs with direct references to a quantity change

Verbs can directly refer to a change in a quantity and its direction, i.e. whether the quantity is increasing or decreasing, when the verb alone contains all the information about the change and the direction and we can therefore distinguish between verbs for positive and negative changes in quantities. For example, *gain, increase,* and *add* are verbs for positive changes, while *lose, decrease,* and *leak* are associated with negative changes.[7] Some verbs belonging to this class also allow prepositional phrase as a complement, which is restricted to the particular direction of change indicated by the verb itself (e.g. 'add to' vs. *'add from').

(34) The *brick* LOSES *heat* to the *room*.
(35) The *temperature of the water* is INCREASING.
(36) The *brick* GIVES OFF *heat*.

Some otherwise 'neutral' verbs can also fall into this class if they use specific particles to indicate a change in a quantity, as in (36).

2.5.2 Verbs with directional prepositional phrases

Verbs associated with Transfer and Motion events do not contain a direct reference to changes in quantity. For example, verbs like *flow* or *move* indicate a transfer of something between two physical or conceptual locations, but they do not contain information about the actual direction of the change. Instead, this information is provided by directional prepositional phrases attached to the verb. The description of the transfer can be complete when both the source and the destination are identified by prepositional phrases, as in (37), or partial when only one of the directional prepositional phrases is attached, as in (38) and (39).

(37) Heat is *transferred* FROM inside the house TO the outdoors.
(38) Energy is *moved* TO a new location.
(39) The fan *moves* heat away FROM the processor.

2.5.3 Verbs in combination with quantity-specific adverbs

Quantity-specific adverbs can determine the change in a quantity in conjunction with a quantity-neutral verb. Analogous to verbs with direct reference to a quantity change, the combination of verbs and quantity-specific adverbs can be associated with a decrease in a quantity, as in (40) or with an increase, as in (41).

---

[7] Another distinction could be made between verbs that can only used with extensive quantities. For example, heat can be *added*, while temperature cannot.

(40) The *glass* is COOLING FASTER.
(41) The *molecules* are MOVING FASTER.
(42) The *substance* DISSOLVES FASTER.

Similar to the interpretation of the quantity type from verb/adverb combinations, there are cases in which the same adverb can refer to an increase or a decrease, depending on the verb with which it is used. For example, in the context of (41), the adverb 'faster' would indicate a positive change in the velocity of the molecules, while in (42) it will indicate an increase in the rate at which a substance dissolves.

2.5.4 Nouns with direct references to change

Nouns provide another way of describing changes in physical quantities. They can be divided into similar classes as verbs, i.e. nouns with direct references to a change in a quantity, and nouns that use directional prepositional phrases.

(43) The INCREASE in *temperature* is significant.
(44) The DECREASE in *pressure* caused a failure.

Nouns can directly refer to a change in a quantity, and analogous to section 2.5.1 they can be divided into nouns that refer to positive changes, as in (43), and negative changes, as in (44).

2.5.5 Nouns with directional prepositional phrases

Similar to verbs of the Transfer and Motion domain, the corresponding nouns will also need directional prepositional phrases to describe changes in a quantity. Again, the information about the transfer can be complete, as in (45) or partial as in (46).

(44) The *flow* of oxygen FROM the tank TO the capsule is blocked.
(45) The *transfer* of heat TO the kettle has been completed.

2.6 Summary

Chapter 5 of this dissertation describes the design of a controlled language for writing natural language descriptions of physical phenomena. One important aspect in the development of such a language is the goal to eliminate (or at least greatly reduce) any possible syntactic and semantic ambiguity. The identification of patterns used for references to continuous parameters in natural language is an essential part of the

semantic interpretation process, which must include the detection of directly referenced quantities as well as indirect references.

Furthermore, the ability to transform sentences with quantity-specific adjectives and adverbs into an equivalent form that only uses generic, quantity-neutral adjectives and adverbs together with a direct reference to a quantity type provides us with a powerful mechanism for creating canonical references to physical quantities. While a quantity-specific form is more interesting for humans to read, the latter version is easier to process for two reasons:

(a) It explicitly mentions the quantity type. A simple lookup on the noun phrase can determine whether it includes a reference to a known quantity type; it is not necessary to interpret a quantity-specific adjective or adverb to identify the associated quantity type.

(b) Knowledge about a small set of quantity-neutral adjectives and adverbs will be sufficient for the interpretation of comparisons between quantities or changes in a quantity.

However, implicit references to quantities as well as quantity-specific adjectives and adverbs are important, because they are beneficial for the habitability of the controlled language. People are accustomed to the use of these constructs and might consider a grammar that only allows explicit references and generic adjectives as awkward. Our analysis of quantity-specific and quantity-neutral adjectives and adverbs has been done from a semantic perspective. Even more, it is primarily specific to the semantics of Qualitative Process Theory.
Huddleston's discussion of comparisons distinguishes between asymmetric and symmetric comparative items, as well as differentiated ('A is different from B.') and undifferentiated ('A and B are different.') forms of comparison (Huddleston, 1971, 1984). Asymmetric comparative items are always differentiated constructions. Huddleston also discusses a form of comparison that uses two variables, as demonstrated in the sentence 'The river is as wide as the lake is deep'. The interesting fact about this sentence is that it refers to two continuous parameters, associated with two different entities – the depth of the lake and the width of the river. The comparison between the two quantities is made possible by the fact that they are compared along the same dimension (length).

Many quantity-specific adjectives and adverbs form opposing pairs for the same quantity type along a single dimension. For example, 'tall' is the opposite of 'short' for the quantity type 'height', and 'wide' the opposite of 'narrow' for the quantity type 'width' (see Bierwisch (1967, 1989) and Kennedy (2001) for a detailed analysis of polar adjectives). For certain quantity types we can identify not just a single opposing

pair but a set of quantity-specific adjectives. For the quantity type ‘temperature’ we can find adjectives such as ‘warm’, ‘cool’, ‘tepid’, and variations such as ‘lukewarm’ as references besides just ‘hot’ and ‘cold’. It is an interesting question to speculate why this variety of quantity-specific adjectives exists for some quantity types but not for others. Frequent use or familiarity with the concept ‘temperature’ cannot explain this fact alone. Quantity types of the length dimensions such as length, height, depth, width, or distance are also frequently used, yet they do not show the same variety of quantity-specific adjectives as the quantity type ‘temperature’.

Furthermore, (Kennedy & McNally, 1999) distinguish between two different types of scales for abstract representations of measurement, i.e. scales. A closed scale has minimal and maximal elements, while an open scale does not. Gradable adjectives like ‘full’/’empty’ or ‘open’/’closed’ possess a gradable property with minimal and maximal values, while adjectives like ‘hot’/’cold’ or ‘tall’/’short’ do not. A glass of water cannot be fuller than full, while (at least theoretically) there is always a temperature higher than the current one. (Kennedy, 2000) also notes that ‘proportional modifiers’ such as ‘completely’ or ‘partially’ are only acceptable for adjectives with a closed scale. For example, a glass can be completely empty, but a person cannot be partially tall. This suggests that quantity-specific adjectives can be subdivided into open-scale and closed-scale quantity-specific adjectives. Although this distinction is not important for determining the quantity type itself, the information about the scale is useful for other purposes such as determining the range of values for a quantity or establishing limit points.

Another interesting aspect is the number of ways in which information about physical quantities can be expressed, and why certain comparative constructs are preferred over others. Intuitively, there seems to be a preference for compact quantity-specific comparatives over the more explicit quantity-neutral constructs. However, for instructional purposes, the more elaborate quantity-neutral form might be favorable, since it mentions the quantity type directly and does not rely on the interpretation of the comparative.

Chapter 3

QP constituents in Natural Language

Natural language contains abundant information about continuous parameters, as the previous has illustrated. In many cases, this data is not just an isolated description about a physical quantity but part of a reference to other constituents of Qualitative Process Theory that combine information about physical quantities. After a brief review of Qualitative Process Theory, this chapter investigates the various forms and syntactic patterns in which information about QP constituents can appear in natural language.

In QP Theory, physical changes in continuous properties are caused by *physical processes*. Examples of physical processes include flows of various substances (e.g., heat, liquid, gas), phase changes (e.g. boiling, freezing), and motion. Ontologically, physical processes serve as the mechanisms of physical causality: All naturally occurring changes and many of the indirect effects of the actions of agents are ultimately caused by the activity of one or more physical processes. Instances of physical processes exist when an appropriate configuration of *participants* occurs. Such process instances are *active* over any span of time for which their *conditions* hold. When a process instance is active, its *consequences* hold. For example, two entities (i.e., having the continuous property *heat*) that are thermally in contact give rise to two instances of heat flow, one in each potential direction. Whether or not either of these is active depends in turn on the relative temperatures between the two bodies.

The consequences of a physical process are of three types. First, there are *direct influences* that represent the direct effects that a physical process has on the world. For example, heat flow causes the heat of the source of the flow to decrease while increasing the heat of the destination. Second, there are other dynamical properties defined, including new parameters and causal laws, which describe how changes propagate through continuous properties. For example, the rate at which heat flows is a continuous property, and it is determined by the difference between the temperatures. Third, other properties that hold while the process is occurring, such as appearance information, can be consequences. In everyday boiling, for instance, one typically sees bubbles.

3.1 Patterns for constituents of physical processes

In the following section, we look at the different forms in which information about the constituents of physical processes can appear in natural language. The examples used in this analysis are taken from our corpus material, which included a popular science book on solar energy (Buckley, 1979) as well as textbook chapters on heat and temperature (Maton et al., 1994; Moran & Morgan, 1994).

3.1.1 Process names

Process names are nouns such as ‘evaporation’ or compound nouns like ‘heat flow’ for descriptions of physical phenomena. In many instances, these nouns are omitted from the description and are referred to indirectly, usually by a verb or by a combination of nouns and verbs.

(1) The heat flows from the hot brick to the cool room

For example, sentence 1 does not explicitly mention a physical process. However, the name of the process (‘heat flow’) can be reconstructed from the subject of the sentence (‘heat’) and the base form of the main verb (‘flow’).

3.1.2 Sub-/Superclasses of processes

Sub- and superclasses of a physical process are ontological extensions, i.e. specializations or generalizations of an existing type of process. For example, convection heat flow is a specialization of a generic heat flow process. A flow process is a generalization of the heat flow process. Subclasses inherit all the properties of their superclass.

(2) There are four important types of thermal resistance, corresponding to each of the four important ways in which heat moves in solar heating systems; they are: 1. Conduction, 2. Convection, 3. Radiation, 4. Transport.

In sentence 2, the convection heat flow process would inherit the participant, condition, and consequences slots of the generic heat flow process. Sub- and superclasses are usually introduced by phrases like ‘type of’ or ‘kind of’, as in “A is a kind of B” or “Types of A are B, C, and D.”

3.1.3 Participants

Participants are the primary actors and entities in physical processes and are typically encoded as noun phrases in the process descriptions. For example, the nouns ‘brick’

and ‘room’ in (3) should be considered participants of a heat flow process. Although ‘heat’ is also a noun, it denotes a quantity type and refers to a particular property of ‘brick’ and ‘room’. Similarly, the ‘pitcher’, the ‘glass’ and the ‘water’ in sentence 4 are participants, while ‘volume’ is a quantity type.[1]

(3) A hot brick loses heat to a cool room.
(4) When you pour water out of a pitcher into a glass, volume flows from the pitcher to the glass.

In combination, the name of a participant and its associated quantity type are sufficient information to describe a physical quantity, as illustrated in the previous chapter.

3.1.4 Conditions

We distinguish between two types of conditions that are relevant in the context of physical processes. *Preconditions* need to be met for an instance of a process to occur, e.g. two thermal objects must be heat-aligned for conduction heat flow. *Quantity conditions* determine when an instance of a process becomes active (or inactive), e.g. a difference in temperature between two objects is a quantity condition for a heat flow process. Even if the objects are in contact and all preconditions are met, the actual heat flow between them will only take place when their temperature differs. Conditions often involve a comparison of quantities, such as ordinal relations, or an explicit reference that it is the cause for an underlying physical process, as illustrated by the following sentences.

(5) Heat flows from one place to another because the temperature of the two places is different.
(6) The flow from the cylinder to the ground stops, because the pipe is blocked.

Sentence 5 shows an ordinal relation that functions as a quantity condition. The heat flow process is active because (and as long as) the difference between the two locations is different. Conditions can also be explicitly mentioned by the use of certain verbs that refer to the beginning or end of an action. In (6), the flow process becomes inactive because of the blocking condition.

---

[1] As it has been noted in chapter 3, ‘volume’ stands in for the actual substance that flows from the glass to the pitcher, similar way to a transfer of heat energy as in ‘Heat flows from the stove to the kettle.’

Quantity conditions use explicit causative patterns, such as `If <Condition>, <Process>`, `<Process> because <Condition>`, `<Condition> causes <Process>`, or variations of these, while preconditions show no clear patterns.

3.1.5 Ordinal Relations

Ordinal relations express relative magnitude relationships between quantities, such that a quantity associated with one participant is more, less, equal to, or different from a quantity of the same type associated with another participant. Ordinal relations can simultaneously function as the activating conditions for a process, i.e. the inequality between the quantities associated with two participants enables the process to become active.

(7) If two cans having different depths are connected by a tube, volume will always flow toward the depth that is lower.
(8) Because the water is warmer than the ice, heat moves from the water to the ice.

Ordinal relations in natural language are typically based on comparisons, in which concrete numbers or some other type of comparative construction (cf. Huddleston, 1984) is used.[2]

The ordinal relation in sentence 7, the difference in depth, does not specify which of the two cans has the greater depth; all it says is that the depths are different. The information gathered from such underspecified ordinal relations might be updated later. The pattern for differences in quantities explicitly mentions the quantity type. If the entities are missing, they have to be determined from the context of the sentence. The pattern also leaves open which of the combined quantities is greater, i.e. the ordinal relation between the quantities is not specified.
In sentence 8, the quantity type (temperature) is mentioned only implicitly by the adjective ‘warmer’. To make this comparison work, the reader has to know that ‘warm’ is a quantity-specific adjective and associated with the quantity type ‘temperature’.[3] Figure 3.1 shows the patterns that are commonly used in natural language for expressing ordinal relationships.

---

[2] A special form of comparison uses implicit references to points or classes. For example, ‘The man is tall.’ compares a particular man against the average height of a group of men or all men, depending on the context of the sentence. In this case, the ordinal relationship is not between two instances but between an instance and a collection of individuals.
[3] The association of the adjectives ‘warm’ and ‘cold’ with the concept ‘temperature’ is a learned fact, since there is no morphological connection between these words (unlike, for example, ‘dense’ and ‘density’). As an alternative, the Transformation hypothesis can be used to rewrite the sentence by

Related to ordinal relations denoted as a difference are *combined quantities* that are the result of the (arithmetic) difference between (or sum of) the two individual quantities. For example, in a sentence like ‘The heat flows because of the difference in depth between the room and the outdoors’ the depth difference can be treated as a

OR1: Difference between quantities, noun

```
<QType>     DIFF/N [between <Entity1> and <Entity2>]
DIFF/N  in <QType> [between <Entity1> and <Entity2>]
```

Example: "A difference in depth causes the water to flow from the can to the cylinder."

OR2: Difference between quantities, adjective

```
<QType> [<Entity1>] [and <Entity2>] DIFF/ADJ
[<Entity1>] [and <Entity2>] DIFF/ADJ <QType>
```

Example: "The temperature of the room and the outside is different."

OR3: Quantity-neutral comparisons

```
<Entity1> <COMP/Qneutral> <QType> than <Entity2>
<Entity1> <Stuff> VP <COMP/Qneutral> than <Entity2>
<Quantity1> <COMP/Qneutral> than <Quantity2>
```

Example: "The big pan of water has more heat than the hot little stone."

OR4: Quantity-specific comparisons

```
<Entity1> <COMP/Qspecific> than <Entity2>
```

Example: "The stone is hotter than the water."

OR5: Quantity-specific adjective combinations

```
<ADJ1/Qspecific> <Entity1>
... <ADJ2/Qspecific> <Entity2>
```

Example: "the cold plate" ... "the hot food"

Figure 3.1: Patterns for Ordinal Relations

quantity by itself. The distinction between combined quantities and ordinal relations in natural language is largely of syntactic nature. As exemplified by (9), combined quantities often use nouns like ‘difference’, while the adjective ‘different’ is used for (underspecified) ordinal relations.

---

replacing the quantity-specific construct with the quantity-neutral equivalent ‘greater temperature’ to make the quantity type explicit.

(9) The difference in temperature between the room and the outside causes the heat to flow.

Only quantities of the same quantity type can be combined, since looking for the difference between the temperature of a brick and the water level in a can would not make any sense.

3.1.6 Miscellaneous Antecedent Relations

Miscellaneous antecedent relations are often found in sentences that contain scenario information such as connections between participants or containment relations. These various pieces of information can also include also preconditions of the process. For example, the fact that two participants are connected via a path or that a can contains a particular liquid are typical preconditions for a flow process, even if they are no explicitly labeled as a condition.

(10) Two cans are connected by a tube.
(11) The window separates the outside air from the air in the room.

Miscellaneous antecedent relations are often found in sentences that use verbs to describe static relationships such as connections between participants, e.g. ‘A is connected to B’ or ‘A and B are conjoint at C’, or containment relations, such as ‘A is in B’ or ‘A contains B’.

Although there are no distinct patterns to extract this type of information directly from the description, data tied to events that serve as conditions for a physical process can be treated as miscellaneous antecedent relations. For example, if the presence of a heat path between two thermal objects is a condition for a heat flow process, the fact that the path connects the two entities can be treated as a miscellaneous antecedent relation.

3.1.7 Direct Influences

Direct influences constrain the ways in which quantities can change. For example, the amount of water that is transferred from one location to another in a volume flow process during some time interval is characterized by the flow rate. The flow rate directly influences the quantity of water at the source and the destination of the flow process. An overt pattern for direct influences would be ‘A is increasing at the rate of B’ or ‘P decreases A by B’. However, explicit information about direct influences on quantities is usually sparse. With a few exceptions, the corpus material analyzed in (Kuehne & Forbus, 2002) contained almost no explicit references to rates.

More often, we will find implicit references for the change of a quantity over time in sentences that contain time-specific references (e.g. adverbs such as ‘fast’ or ‘slowly’) or verb phrases that express dynamic changes (e.g. ‘drives out’, ‘flows from/to’). These references do not explicitly express the rate of change or even mention that a quantity is increasing or decreasing. Instead, a semantic interpretation of the verbs and adverbs is needed to extract this implicit information.

DI1: Transfer between quantities (active voice)

```
<QType> <Change> [from <Entity1>]
                 [to <Entity2>] [via <Path>]
```

Example: “Heat flows from hot things to cold things.”

DI2: Transfer between quantities (passive voice)

```
<QType> <Change> [by <Agent>] [from <Entity1>]
                 [to <Entity2>] [via <Path>]
```

Example: “Volume is transferred from the can to the ground.”

DI3: Explicitly mentioned transfer event

```
<Change> <QType> [from <Entity1>] [to <Entity2>]
```

Example: “The flow of heat from the hot brick to the cool ground ...”

DI4: Quantity change in object (active voice)

```
<Agent> <Change> <QType> [from <Entity1>]
                         [to<Entity2>] [<Path>]
```

Example: “A can of water leaks volume from a hole to the ground.”

DI5: Quantity change in object (passive voice)

```
<QType> <Change>    by <Agent>
<QType> <PosChange> to/by <Agent> [from <Entity>]
<QType> <NegChange> from/by <Agent> [to <Entity>]
```

Example: “Heat is gained by the room.”

Figure 3.2: Patterns for Direct Influences

(12) How quickly the ice melts will measure how much heat is flowing through the bar from the coffee.

(13) Heat flows from the ground to the air.

The common patterns for direct influences are shown in Figure 3.2.

3.1.8 Indirect Influences

Indirect influences express qualitative proportionalities between quantities. They describe how changes in one quantity can cause changes in another. For example, the heat and the temperature of a thermal object are usually qualitatively proportional; all else being equal, the more heat the object has, the higher its temperature is.

(14) The bigger the thermal resistance, the harder it is for heat to flow, since the resistance to the flow of heat is increased.
(15) The larger the surface area is, the more convection heat is lost from the surface.
(16) The flowrate also depends on the area of the heat-flow path.
(17) The faster an object moves, the more kinetic energy it has.
(18) As the temperature rises, the fluid expands.

There are a number of distinctive patterns for indirect influences. For example, one is a comparison pattern that uses a the-the construct plus comparatives, as in 'The *x*-er A, the *y*-er B'. Another pattern connects two parameters by using the causal verbs such as 'depends on', 'causes' or 'influences' (Wolff, 2003; Wolff, Song, & Driscoll, 2002). The causal verbs used in indirect influence pattern differ in specificity from verbs used in direct influence patterns. The verb 'increase' in used to describe the direct influence in 'The flow increases the amount in the tank.'[4] is much more specific than the verb 'affects' in "The area of the path affects the volume flow rate."

Similar to direct influences, there are many instances of indirect influence where we can only find implicit references. Sentences 17 and 18 are examples where one of the quantities is only implicitly referenced. In sentence 17, the adjective 'faster' and the verb 'move' refer to the velocity of the object. In (18) the verb 'expand' stands for an increase in volume of the fluid. The common patterns for indirect influences are shown in Figure 3.3.

[4] This sentence follows the DI4 pattern. The process itself acts as an agent and does not explicitly mention the flow rate as the actual influence on the amount of water.

II1: THE x-er/THE y-er

```
THE <Comparative1> <Quantity1> [<Change1>],
  THE <Comparative2> <Quantity2> [<Change2>].
```

Example: "The larger the surface area is, the more heat is lost from the surface."

II2: AS x,y

```
AS <Quantity1> <Change1>, <Quantity2> <Change2>.
<Quantity1> <Change1>, AS <Quantity2> <Change2>.
```

Example: "As the volume of the gas increases, the density of the gas decreases."

II3: WHEN x,y

```
<Quantity1> <Change1>, WHEN <Quantity2> <Change2>
WHEN <Quantity1> <Change1>, <Quantity2> <Change2>
```

Example: "The liquid in a thermometer expands when it is heated."

II4: Verb-based Patterns

```
<Quantity1> [<Entity1>] DEPENDS ON <Quantity2> [<Entity2>]
```

Example: "The amount of heat produced depends on the amount of motion".

```
<Quantity1> [Sign] AFFECTS <Quantity2>
<Quantity1> AFFECTS <Quantity2> [Sign]
```

Example: "The area of the path affects the volume flow rate."

```
<Quantity1> [Sign] INFLUENCES <Quantity2>
<Quantity1> INFLUENCES <Quantity2> [Sign]
```

Example: "The speed of the airflow affects how quickly the heat flows."

```
<Change1> <Quantity1> CAUSES <Change2> <Quantity2>
```

Example: "Heat gain causes air temperature to rise."

Figure 3.3: Patterns for Indirect Influences

3.1.9 Miscellaneous Consequence Relations

All consequences of a process other than direct and indirect influences can be classified as miscellaneous consequence relations. For example, the effects that can be observed while a physical process is active can be treated miscellaneous consequence relations.

(19) The onset of boiling is marked by bubbles.
(20) The leakage of water from the pipe renders the ground impassable.

Analogous to miscellaneous antecedent relations, the patterns for this type of information appear to be highly content-specific and are hard to characterize in general terms. Information tied to direct and indirect influences or explicitly marked as a consequence of the physical process is treated as miscellaneous consequences, e.g. the fact that the constrained quantity in an indirect influence relation might also function as the destination of a transfer event.

3.2 Landmarks and limit points

Limit points play an important role in reasoning about physical quantities. They determine the points where important changes happen, i.e. certain physical properties occur or objects and processes come into existence (or cease to exist). For example, water changes its state at the boiling point from a liquid into steam.

L1: Action at a point

```
<Point> <Condition>: <Action>
```

Example: "At some point the depth gets so high that the water flows out of the can"

L2: Quantity at a point

```
<Quantity> <VP> AT <Point>
WHILE/DURING <Action>, <Quantity> <VP> AT <Point>
WHEN <Quantity> <VP> <Point>, <Action>
<Action> WHEN <Quantity> <VP> <Point>
```

Example: "The temperature of the water remains at 150 degrees"

L3: Conditional for points and intervals

```
<Conditional> <Quantity> <Point>, <Action>
<Conditional> <Quantity> <Interval>, <Action>
```

Example: "If its depth is above the hole, the fluid leaks out."

L4: Labeling

```
<Value> is <Point>
<Point> is <Value>
```

Example: "The freezing point of water is 0 degrees Celsius"

Figure 3.4: Patterns for landmarks and limit points

Limit points do not have to have a fixed value along their dimension. Although the temperature of the boiling point of water is commonly stated as 212 degrees Fahrenheit, the actual boiling point temperature depends on the environmental conditions such as the pressure (e.g. the altitude at which the boiling process takes place) as well as on other substances that might be dissolved in the water.

The use of landmarks provides a convenient way of labeling specific values of limit points. Staying with the temperature domain, the temperature of the water in a kettle five minutes into a heating process could be labeled as a landmark. Important limit points are, of course, the boiling point and the freezing point of the water. Figure 3.4 shows the patterns used in our corpus for expressing information related to landmarks and limit points.

3.3 Summary

The knowledge that certain QP-related information appears in a particular form allows us to exploit these patterns for parsing and interpreting natural language descriptions. The patterns we have identified for ordinal relations as well as for direct and indirect influences provide the basis for the design of a controlled language in chapter 5 and for the semantic interpretation process in chapter 6.

The constructs of QP Theory analyzed in this chapter can be recast in terms of frame-oriented data structures that are in part inspired by Minsky's notion of frames (Minsky, 1975). In the next chapter we introduce a collection of frame structures as an intermediate representational layer for capturing information about the constituent of physical processes. These frames provide the link between the natural language input and an internal representational form that can be processed by other programs.

Chapter 4

QP Frames – a link between Natural Language and Qualitative Process Theory

The previous two chapters have shown that understanding and interpreting descriptions of physical processes must connect fundamentals of our conceptual structure to their realizations in linguistic forms, and thus must draw upon both insights about language and about conceptual structure.

In this chapter we recast the constituents of QP Theory as a set of specialized representations that link QP-related information found in the natural language input to the semantics of QP Theory. This representation layer connects the lexical and syntactic analysis (discussed in chapter 5) with the semantic interpretation process (chapter 6). Furthermore, these representations are compatible to FrameNet (Fillmore et al., 2001), a large-scale project in computational linguistics.

4.1 Frame Semantics

FrameNet set out to develop broad systems that capture these aspects of word meaning and linguistic constructions in terms of *frame semantics* (Fillmore & Atkins, 1994; Minsky, 1975). In frame semantics, meaning is expressed in terms of systems of structured representations, or *frames*, which provide the links between words and conceptual structures (Petruck, 1996). Lexical items are linked to frames as such that they highlight a particular frame. For example, the occurrence of the noun 'growth' might evoke the `Expansion` frame in the Space domain, while the verb 'push' activates the `Cause-to-move` frame in the Motion domain.

The participants, props, and other conceptual roles involved in a frame are called *frame elements*. Frame elements are linked to parts of a text and have associated with them inferences that provide meaning (Fillmore & Atkins, 1994).[1] For example, the

[1] Similar ideas can be found in other representation systems for NL semantics. For example, Talmy uses semantic elements such as motion, path, figure, ground, or manner in lexicalization patterns (Talmy, 2000). However, Talmy is more interested in universal patterns that hold across languages, less in the semantic-to-surface associations (as in the FrameNet project).

`Motion` frame includes frame elements for the agent, the theme (i.e. the object acted on), the source, the path, the goal (i.e. the destination) and many more. Not all frame elements of a frame are always present. Many of them are optional and are used in combinations to express specific grammatical realizations of a frame. Although frame elements are similar to thematic roles and case roles, it is important to note that frame semantics does not define a universal set of possible frame elements. Frame elements

Motion

Definition:
Some entity (Theme) starts out in one place (Source) and ends up in some other place (Goal), having covered some space between the two (Path). The frames that inherit the general Motion frame add some elaboration to this simple idea. Inheriting frames can add Goal-profiling (arrive, reach), Source-profiling (leave, depart), or Path-profiling (traverse, cross), or aspects of the manner of motion (run, jog) or assumptions about the shape-properties, etc., of any of the places involved (insert, extract).

Frame Elements:

Area:	The setting in which the Theme's movement takes place.
Carrier:	The means of conveyance of the Theme.
Distance:	Any expression, which characterizes the extent of the Motion.
Duration:	The Duration of time for which the Motion takes place.
Goal:	The location the Theme ends up in.
Path:	Reference to (a part of) the ground over which the Theme travels or to a landmark by which the Theme travels.
Source:	The location the Theme occupies initially before its change of location.
Speed:	The Speed at which the Theme moves.
Theme:	The entity that changes location. Note that it is not a self-mover.

Lexical Units
blow.v, coast.v, drift.v, float.v, fly.v, glide.v, go.v, move.v, roll.v, soar.v

Figure 4.1: The FrameNet Motion frame

are specific to the frame they are defined in, and more frame elements can be added to a frame if the underlying grammatical structures require them.

Figures 4.1 and 4.2 show the FrameNet frames for `Motion` and `Fluidic_Motion`. The definitions and the descriptions of the frame elements are those found in the online version[2] of the FrameNet 2 data (Fillmore & Baker, 2001). Note that the two frames share many of their frame elements, such as the `Area`, `Carrier`, `Distance`, `Duration`, `Goal`, `Path`, `Source` and `Speed`.

Fluidic_Motion

Definition:
In this frame, a Fluid moves from a Source to a Goal along a Path or within an Area.

Frame Elements:

Area:	The setting in which the Fluid's movement takes place.
Carrier:	The means of conveyance of the Fluid.
Distance:	The physical extent of the motion of the Fluid.
Duration:	The amount of time for which the Fluid moves.
Flow_unit:	Information about the amount and unitization of the flow.
Fluid:	The entity that changes location and moves in a fluidic way.
Goal:	The location the Fluid ends up.
Path:	The trajectory along with the fluid moves.
Result:	The result of the Fluid moving.
Source:	The location the Fluid occupies initially.
Speed:	The rate at which the Fluid flows.

Lexical Units:
bubble.v, cascade.v, course.v, dribble.v, drip.v, flow.v, gush.v, jet.v, leak.v, ooze.v, percolate.v, purl.v, run.v, rush.v, seep.v, soak.v, spew.v, spill.v, splash.v, spout.v, spurt.v, squirt.v, stream.v, trickle.v

Figure 4.2: The FrameNet Fluidic_Motion frame

The `Motion` frame uses the frame element `Theme` to mark an (not self-moving) object that changes location, while the `Fluidic_Motion` frame provides the frame element

[2] The online version of the FrameNet database is available at `http://www.icsi.berkeley.edu/framenet`

`Fluid` for a similar purpose. Additionally, the `Fluidic_Motion` frame allows the frame elements `Flow_Unit` and `Result` that have no equivalents in the `Motion` frame.
Frames can range from simple patterns and states to highly complex scenarios. Scenarios consist of several scenes and transition states (which are also frames) and information about their temporal ordering and occurrence. For example, a basic physical process frame, whose structure provides the fundamental aspects of physical processes, can have a number of subframes. These subframes are elaborations of their parent frame and describe particular categories of physical processes, with differences in their participants and consequences being the differentia that set them apart. A subframe inherits all the frame elements of its parent frame and might add several new ones. Instances of frames can be combined with other frames to create the frame system describing the meaning of a text. FrameNet provides support for multiple inheritance, frame blending (i.e. the simultaneous activation of two frames), and frame composition (i.e. the definition of scenario as a sequence of scenes) (Johnson et al., 2001).

4.2 QP Frames

We have recast QP Theory as a set of specialized frames structures. These QP frames use a representational scheme that is compatible with the notions of frames and frame elements in FrameNet. The packaging of physical knowledge and principles in QP Theory (inspired in part by Minsky, 1975) suggests a natural alignment with frame semantics. QP frames are intended to capture information about physical processes expressed in natural language text. QP frames form an intermediate representational layer between natural language and information specific to qualitative models, such as CML model fragments (Falkenhainer et al., 1994).

The qualitative causal mathematics of QP Theory is expressed through another collection of frames. In addition to their role in physical process descriptions, these qualitative causal frames can be used for other domains with continuous parameters, such as economics or metaphorical extensions of physical concepts.

The semantic interpretation process described in chapter 6 of this thesis builds QP frames from the information supplied by the parser. Interpretation rules operate over QP frames by merging and analyzing frames structures. The results of the semantic interpreter can be used to build model fragments for qualitative simulators, such as GIZMO (Forbus, 1984) or other qualitative reasoning systems, such as SIMGEN (Forbus & Falkenhainer, 1990). The following section provides an overview of the set of QP frames and illustrates the type of information captured by their frame elements.

4.2.1 The Quantity frame

The central QP frame structure is the Quantity frame, which captures information about continuous parameters. Since physical quantities play a fundamentally important role, every other type of frame will use Quantity frames for one or more of their frame elements. The frame itself defines the following five frame elements:

- **entity** specifies what this property is a property of. Linguistically, this is a unique discourse variable representing a particular entity. Example: “brick32” in “the weight of the brick.”
- **quantityType** specifies the kind of parameter that this is. Example: “temperature” in “the temperature of the water.”
- **quantityValue** specifies the numerical value of the property. This frame element is optional. Example: “3” in “3 liters of water.”
- **quantityUnit** specifies the physical units of the property. This frame element is optional. Example: “kilograms” in “3 kilograms of lead”.
- **signOfDerivative** specifies how the parameter is changing. This frame element is optional. Example: In “The temperature is increasing.” The sign is expressed by the word “increasing” which would be mapped to the value of `Positive`. While syntactic realizations for quantity types, values and units are fairly obvious, the sign of derivative manifests itself in the text many different ways, e.g. `Negative` could appear as “falling” or “decreasing” in the input text.

Only the first two elements, the entity and the quantity type, are necessary to instantiate a Quantity frame. The remaining three elements, the value, its unit, and the sign of the derivative, are optional (see chapter 2). Quantity Frames are the basic representational unit of QP frames and will be used in every other type of QP frame.

4.2.2 The OrdinalRelation frame

Although values and units are often not explicitly stated or even filled in via default, comparative statements about values are common. These are expressed via the OrdinalRelation frame, which has the following three frame elements:

- **quantity1**, **quantity2** specify the two quantities being compared.
- **ordinalReln** specifies the relationship between the values of the quantities.

Ordinal relations provide a useful qualitative notion of value because they often serve as conditions for physical processes and states (e.g., flows occur when a driving parameter is unequal, equilibriums occur when opposing effects are equal). Syntactic realizations of ordinal relations are usually described via explicit comparisons (e.g., ‘Q1 is greater than Q2’) or as some type of comparative construction. One very

common pattern is the use of dimensional adjectives to set up a tacit comparison via dimensional adjectives.

For instance, from the noun phrases ‘hot brick’ and ‘cool room’ one can construct an ordinal relationship involving their temperature due to the meanings of ‘hot’ and ‘cool’. See chapter 2 for a detailed analysis of the syntactic forms found in comparisons.

4.2.3 The Influence frame

The causal relationships between quantities are expressed via a qualitative mathematics that supports partial information about the nature of the connections between them. The basic frame is the Influence frame, whose frame elements are:

- **constrained** specifies the dependent quantity, i.e., the effect.
- **constrainer** specifies the independent quantity, i.e. a proximal cause for the constrained quantity.
- **sign** specifies the direction, which can be positive or negative.

There are two subframes of the Influence frame, DirectInfluence and IndirectInfluence. These two types of frames correspond to the QP Theory primitives `I+/I-` (direct influences) and `Qprop+/Qprop-` (indirect influences) respectively (Forbus, 1984). While the two subframes share the same frame elements, their underlying semantics are different and will be discussed separately.

4.2.3.1 The IndirectInfluence frame

In the IndirectInfluence frame, the constrained quantity is functionally dependent on the Constrainer, and perhaps on other properties as well, with the sign indicating whether the dependence is increasing or decreasing monotonic. This is the weakest distinction that enables changes to be propagated through causal laws.

The syntactic realizations for the Indirect Influence frame are described in chapter 3. For example, the sentence ‘As the air temperature goes up, the relative humidity goes down’ clearly uses a syntactic pattern for indirect influences. The constrained quantity is the ‘relative humidity’, the constrainer would be the ‘air temperature’, and the sign is negative.

4.2.3.2 The DirectInfluence frame

For direct influences, the constrainer can be combined via addition with other constrainers to determine (qualitatively) the derivative of the constrained quantity, and the sign indicates whether it is a positive or negative contribution to that sum.

Syntactic realizations for DirectInfluences are more complex and do not follow strict patterns (chapter 3). In advanced texts, one can find syntactic constructs such as '*The rate of [constrained] depends on [constrainer].*' In everyday texts, explicit discussions of rates are very rare. Instead, DirectInfluences tend to occur in larger-scale patterns, often tied to periphrastic verbs (Wolff, 2003). For example, the sentence '*Most water in the air comes from evaporation.*' uses a DirectInfluence frame, with 'water in the air' as the constrained quantity, the '(rate of) evaporation' as the constrainer, and a positive sign.

4.2.4 The QuantityTransfer Frame

The QuantityTransfer frame captures information about the transfer of some property between two quantities. This kind of transfer is often implied by descriptions of flow and motion events that use the common patterns for Direct Influences. The QuantityTransfer frame consists of the following three frame elements:

- **sourceOfTransfer** is the name of the source quantity participating in the transfer event. This is the quantity that will lose some of its property. Sources are typically indicated by the preposition *from*, as in the phrases such as 'from the brick'.
- **destOfTransfer** is the name of the quantity that will gain some of the transferred property. It is usually referred to by prepositions such as *to, toward*, or *into.* Examples: 'to the ground' or 'into the cylinder'.
- **rateOfTransfer** is the name of the quantity that specifies the transfer rate, i.e. how fast the property is transferred between the source and the destination. As mentioned in the previous chapter, explicit information about rates is rarely found in descriptions of physical processes.

The QuantityTransfer does not specify the transferred quantity type explicitly as a frame element. Instead, the quantity type is part of the Quantity frames specified via the **sourceOfTransfer** and **destOfTransfer** frames. This means that the quantity type for the source and the destination quantities have to match in a QuantityTransfer frame.

Furthermore, the transfer from a source can have multiple destinations, and a transfer to a single destination can come from multiple sources. For example, the water flow from a cylinder can be split into two separate flows by a Y-shaped junction pipe. Furthermore, the two ends of the junction pipe can have different diameters, resulting

in different flow rates between the source and the destination. In this case, two separate QuantityTransfer frames would be used to model this flow to different destinations at different flow rates. Both QuantityTransfer frames would have the same Quantity frames specified as their **sourceOfTransfer** and different Quantity frames as their **destOfTransfer** quantity. The two **rateOfTransfer** quantities associated with each QuantityTransfer frame capture the separate flow rates from the source to the two destinations.[3]

4.2.5 Processes and their occurrences

The PhysicalProcess frame combines information from several QP frames. The four frame elements of the PhysicalProcess frame are:

- **participant** specifies one of the participants in the physical process. For example, in the sentence 'Heat flows from the hot brick to the cool room', the noun phrases 'hot brick' and 'cool room' denote participants in an instance of a heat flow process.
- **condition** specifies one of the conditions under which the process is active. In the sentence 'Heat flows from one place to another because the temperature of the two places is different.' the condition is the difference in temperature values.
- **consequence** specifies one of the direct consequences of the physical process. In the sentence 'Water flooded into the room, because the valve broke.' the liquid flow into the room has an increase in the amount of water in the room as one of its consequences.
- **status** specifies whether a process is active. In the sentence 'The radiator leak was stemmed by shoving a cloth into it.' the verb 'stemmed' suggests that a previously enabled flow is now stopped. The status is *active* when the process is occurring, and *inactive* otherwise.

These frame elements can be directly mapped to the formal models that QP Theory supports. For a process type or instance, the set of participants collectively define the collections of entities it occurs among. The union of the conditions is the set of conjuncts that comprise the necessary and sufficient conditions for it to be active. The set of fillers for the consequences frame elements constitutes its direct consequences.

Noun phrases that serve as the primary actor and object in a sentence are considered participants, e.g., in the sentence 'A hot brick loses heat to a cool room.' the noun phrases 'hot brick' and 'cool room' denote participants. The patterns that indicate

---

[3] The use of two separate QuantityTransfer also allows a zero flow rate for a **rateOfTransfer** quantity if one of the outlets to the corresponding **destOfTransfer** quantity is blocked.

conditions include '[condition] causes [process]', '[process] occurs when [condition]', and '[process] depends on [condition].' These patterns are addressed in more detail in chapter 5 within the context of the controlled language used for describing physical phenomena. For consequences, there are two cases: influences and other consequences. Influences are captured by the DirectInfluence and IndirectInfluence frames. The other consequences can range over almost any physical statement in principle (e.g., appearances, sounds, etc.). These miscellaneous consequences are difficult to characterize in terms of specialized QP frames. The system keeps assertions about miscellaneous consequences as part of the semantic interpretation data (see chapter 7 for a detailed example).

4.3 Integration of QP Frames into the Cyc KB

Combining the FrameNet resources with the information provided by the background knowledge base is highly desirable for producing general semantic interpretations of physical phenomena. For example, general frames could be used for miscellaneous antecedents and consequences that do not fit the specialized QP frames. We use the Cyc knowledge base (Lenat & Guha, 1989) to provide background knowledge for the information about the lexical items of the natural language input and the semantic interpretation process. Unfortunately, the current version of the Cyc knowledge base does not include the FrameNet data. We discuss the possibility of integrating the general FrameNet data with Cyc in more detail in chapter 7. This section illustrates how QP frame structures are incorporated in the Cyc knowledge base.

The QP frame structures are defined as two collections, `QPFrame` and `QPFrameElement`, in the upper ontology. `QPFrame` is a subcollection of the collection `Frame`. `Frame` itself is a subcollection of `Situation`, which covers `Events` and `StaticSituations`. The collection `Frame` could eventually be used for incorporating the actual FrameNet content, making `QPFrames` as specialized subset of `Frames`. Similarly, `QPFrameElement` is defined as a subcollection of `FrameElement`, which is a specialization of `Role`. The actual frames and their frame elements are then defined in terms of these collections.

The following expressions define the Quantity frame and its five frame elements. The frame elements are tied to the frame by using the predicate `usesQPFrameElement`. A frame element can be tied to a number of frames without defining it individually for each frame. For example, the frame element `entity` could be reused in other frames besides the Quantity frame. This mechanism closely corresponds to the way in which frame elements are used in FrameNet.

```
(genls QuantityFrame QPFrame)
(usesQPFrameElement QuantityFrame entity)
(usesQPFrameElement QuantityFrame quantityType)
(usesQPFrameElement QuantityFrame quantityValue)
(usesQPFrameElement QuantityFrame quantityUnit)
(usesQPFrameElement QuantityFrame signOfDerivative)

(isa entity QPFrameElement)
(arg1isa entity QPFrame)
(arg2isa entity Thing)
(arity entity 2)

(isa quantityType QPFrameElement)
(arg1isa quantityType QPFrame)
(arg2isa quantityType PhysicalQuantity)
(arity quantityType 2)

(isa quantityValue QPFrameElement)
(arg1isa quantityValue QPFrame)
(arg2isa quantityValue SubLRealNumber)
(arity quantityValue 2)

(isa quantityUnit QPFrameElement)
(arg1isa quantityUnit QPFrame)
(arg2isa quantityUnit QPUnit)
(arity quantityUnit 2)

(isa signOfDerivative QPFrameElement)
(arg1isa signOfDerivative QPFrame)
(arg2isa signOfDerivative QPSign)
(arity signOfDerivative 2)
```

QP Frames are tied to events and concepts already existing in the knowledge base by the predicate `relatedQPFrame`. Knowing the QP frame associated with a standard Cyc event allows us to look up and assert further information such as the roles indicated by the `supportsRoleInFrame` assertions. For example, the following assertions tie a set of common flow and motion events to the QuantityTransfer frame.

```
(relatedQPFrame Translation-Flow QuantityTransferFrame)
(relatedQPFrame RollingOnASurface QuantityTransferFrame)
(relatedQPFrame Running QuantityTransferFrame)
(relatedQPFrame Walking-Generic QuantityTransferFrame)
(relatedQPFrame MovementEvent QuantityTransferFrame)
```

The `supportsRoleInFrame` predicate allows a mapping from the general Cyc predicates for role relations to the specific roles for a particular QP frame. Although it would be possible have specialized rules for each of these mappings, the use of `supportsRoleInFrame` assertions creates an intermediate layer and allows many-to-

one mappings between standard Cyc role relations and the roles an item plays in a QP frame.

```
(supportsRoleInQPFrame QuantityTransferFrame objectMoving
                       transferredStuff)
(supportsRoleInQPFrame QuantityTransferFrame transferredThing
                       transferredStuff)
(supportsRoleInQPFrame QuantityTransferFrame from-Generic
                       sourceLocOfTransfer)
(supportsRoleInQPFrame QuantityTransferFrame to-Generic
                       destLocOfTransfer)
(supportsRoleInQPFrame QuantityTransferFrame by-Underspecified
                       pathOfTransfer)
(supportsRoleInQPFrame QuantityTransferFrame instrument-Generic
                       pathOfTransfer)
(supportsRoleInQPFrame QuantityTransferFrame doneBy
                       actorOfTransfer)
(supportsRoleInQPFrame QuantityTransferFrame providerOfMotiveForce
                       actorOfTransfer)
```

The specialized role relations are not the predicates for the frame elements but provide information for constructing the frame element information. A good example is the **sourceOfTransfer** frame element of the QuantityTransfer frame. The general semantic interpretation process (chapter 6) might produce an expression like `(from-generic flow123 can456)`. The entity 'can456' cannot be used as a filler for the frame element **sourceOfTransfer**, because it expects a quantity instead of an entity. To construct the quantity frame to be used with the **sourceOfTransfer** frame element the location information from the 'from-generic' expression needs to be combined with information about a quantity type participating in the same transfer event.

4.4 Capturing NL information in QP frames

This section illustrates how the QP-specific information included in sentences can be captured in QP frames, and how a relatively simple process model can be generated from the frame data.

4.4.1 Example 1

Sentence 1 describes a simple heat flow process between two entities. Besides mentioning the source and the destination of the flow, it also provides information why the heat flows between them.

(1) The heat flows from the brick to the ground, because the brick has a higher temperature than the ground.

The sentence includes the following information:

- A flow event
- Two entities, a brick and the ground.
- Two quantity types, heat and temperature.
- The source and the destination quantities of the transfer, the heat of the brick and the ground, respectively.
- An ordinal relationship between the temperatures of the brick and the ground.
- A causal relationship between the temperature difference and the flow of heat

The first part of the sentences describes the flow event that transfers heat from one location to another. To capture this information, a QuantityTransfer frame (QT1) is used, which in turn requires three Quantity frames to be instantiated for the source (Q1), the destination (Q2), and the rate of the flow (Q3). The data of the QuantityTransfer frame can then be used to create two DirectInfluence frames DI1 and DI2, i.e. for the decrease of heat at the source and the increase of heat at the destination. Figure 4.3 provides an overview of the six QP frames.

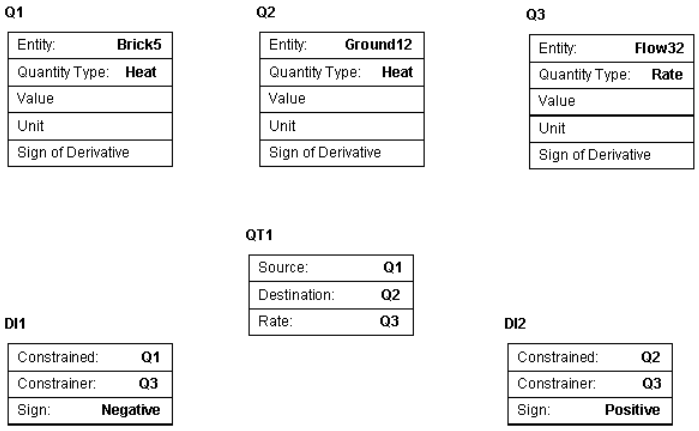

Figure 4.3: QP frames for a quantity transfer – Example 1

The second part of the sentence describes an ordinal relation between the two quantities, the temperature of the brick and the temperature of the ground. Two new Quantity frames Q4 and Q5 are instantiated for the same entities as in Q1 and Q2. The comparison in this sentence part identifies the ordinal relationship between the two

quantities and leads to the creation of the OrdinalRelation frame OR1. Figure 4.4 shows the resulting frame structure for this part of sentence 1.

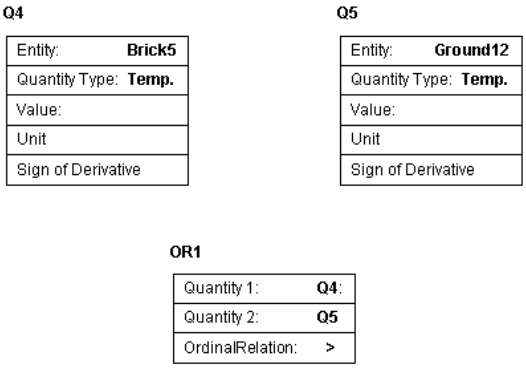

Figure 4.4: QP frames for an ordinal relation – Example 1

The information captured in the frame structures can now be combined into a PhysicalProcess frame. The two entities are treated as the participants of the flow process. The sentence itself identifies the comparison between the temperature of the brick and the ground as the cause for the flow. Therefore, the OrdinalRelation frame OR1 will serve as a condition for the flow. Finally, the QuantityTransfer frame QT1 and the DirectInfluence frames DI1 and DI2 are consequences of the flow process, since the transfer only takes place as long as the process is active. Figure 4.5 illustrates how the captured information contributes to the construction of the PhysicalProcess frame.

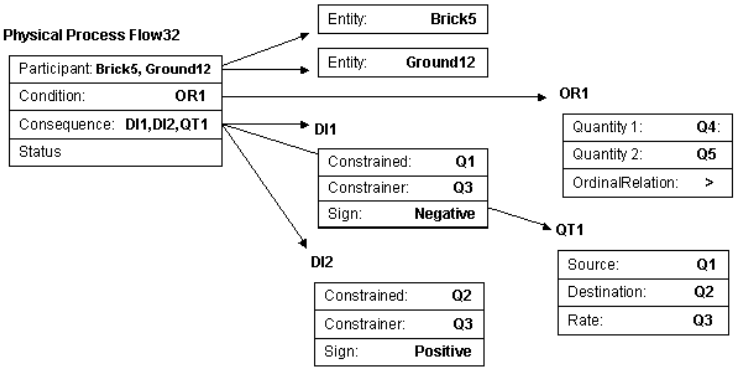

Figure 4.5: Process frame – Example 1

The PhysicalProcess frame can be transformed into other representation forms for further processing. Figure 4.6 shows an instantiated model in CML (Falkenhainer et al., 1994) that is based on the information contained in the PhysicalProcess frame.

```
(defModelFragment flow32
 :subclass-of (Translation-Flow)
 :participants
  ((brick5 :type Brick)
   (ground12 :type Ground))
 :conditions
  ((> (temperature brick5)
      (temperature ground12)))
 :quantities
  ((rate))
 :consequences
  ((fromLocation flow32 brick5)
   (toLocation flow32 ground12)
   (I- (heat brick5) (rate flow32))
   (I+ (heat ground12) (rate flow32)))
```

Figure 4.6: Process model - Example 1

These descriptions can be used as input to a generalization process that distills CML model fragments from a collection of individual models. We address this issue in more detail as future work in chapter 8.

4.4.2 Example 2

The second example illustrates how information about direct and indirect influences can be extracted from a single sentence.

(2) The greater the thermal resistance of the isolation [is], the less heat flows from the room to the outdoors.

Sentence 2 contains the following QP-related pieces of information:

- A flow event.
- Three entities, the room, the outdoors, and the insulation.
- Two quantity types, heat and thermal resistance.
- The source and the destination of the transfer, the room and the outdoors.
- A causal relationship between the thermal resistance of the insulation and the flow of heat.

The transfer of heat between the room and the outdoors is captured by a set of QP frames similar to those shown in Figure 4.3. Again, the heat of the room is the source quantity of the transfer, while the heat of the outdoors is the destination quantity. The transfer uses the unspecified rate as an internal quantity of the heat flow process.

The syntactic form of (2) matches to the THE/THE pattern for indirect influences. The first part of the sentence identifies the constrainer, while the second part of the sentence contains information about the constrained quantity. In this example, the flow of heat is constrained by the thermal resistance of the insulation. The comparatives ‘greater’ and ‘less’ determine the negative sign for the indirect influence, i.e. the flow will get less with greater resistance. The resulting frame structure is shown in Figure 4.7.

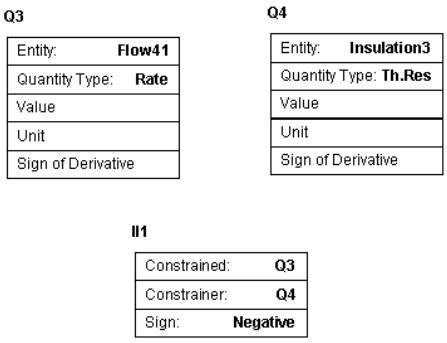

Figure 4.7: QP frames for indirect influences – Example 2

The resulting process model is shown in Figure 4.8. Since the sentence does not mention any conditions for the flow or information about the status, the two corresponding slot are not filled. However, they might be updated at a later time by information that the insulation has to be a heat path between the room and the outdoors to enable the heat to flow. Figure 4.9 shows the information about the flow process as an individual CML model.

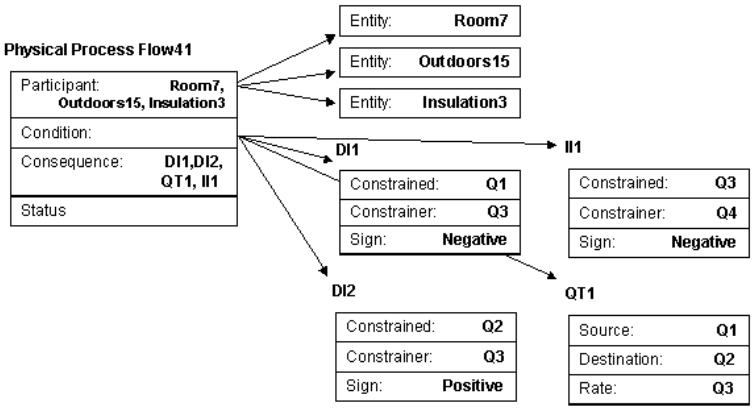

Figure 4.8: Process frame – Example 2

```
(defModelFragment flow41
 :subclass-of (Translation-Flow)
 :participants
  ((room7 :type Brick)
   (outdoors15 :type Ground)
   (insulation3 :type BuildingInsulation)
 :quantities
  ((rate))
 :consequences
  ((fromLocation flow41 room7)
   (toLocation flow41 outdoors15)
   (I- (heat room7) (rate flow41))
   (I+ (heat outdoors15) (rate flow41))
   (Qprop- (rate flow41) (thermal-resistance insulation3)))
```

Figure 4.9: Process model - Example 2

4.5 Summary

In this chapter we have illustrated how QP Theory can be recast as a set of specialized frame structures that provide an extension to FrameNet, and how these QP frame structures can be integrated into the Cyc ontology. Although the FrameNet data has not yet been included in the Cyc knowledge base, this integration would be highly desirable, since FrameNet provides valuable resources to complement the semantic information associated with concepts in Cyc. While the Cyc knowledge base provides a broad ontological foundation, the FrameNet data will add deep, fine-grained semantic information for events and individual concepts. The fact that Cyc's underlying representation language originated from frame-based systems, as evidenced by the use of slots and units in (Lenat & Guha, 1989), should facilitate this integration. Chapter 7 discusses different strategies for an integration of FrameNet with Cyc.

QP frames, as a specialized extension to FrameNet, provide an intermediate representational layer between the actual natural language input and the final representations that can be used in reasoning. Clearly, a successful use of QP frame structures depends on support by the parsing step and the semantic interpretation process. The parser can facilitate the construction of frames by identifying possible candidates for frame elements, which are then evaluated by a subsequent semantic analysis.

The following chapter discusses the design of a restricted input language that we use to describe information about physical processes. This language supports the syntactic patterns identified in chapter 3 and facilitates the construction of QP frames.

Chapter 5

QRG Controlled English – a controlled language for descriptions of physical phenomena

A common problem that natural language processing systems have to cope with is the trade-off between *ambiguity* and *expressiveness* of the language used in the source text. Unrestricted natural language is full of ambiguity, even when the context in which it is used may provide some constraints. Ambiguity can arise from word meanings, e.g. the polysemy of individual words or the interpretation of word compounds, and from grammatical constructs, e.g. multiple interpretations of a sentence based on different prepositional phrase attachments. Sentences like 'Fruit flies like bananas' or 'I saw the man on the hill with a telescope' are classic examples that illustrate the ambiguity of natural language.

The use of a controlled language can reduce ambiguity by restricting the grammar and the lexicon. Controlled languages have a long history that predates the fields of computational linguistics and natural language understanding (C. K. Ogden, 1933, 1937) and have found applications in technical domains such as maintenance of vehicles and machinery. More recently, controlled languages were used for the preparation of technical documentation (Almquist & Sagvall Hein, 1996; Wojcik et al., 1998), logic representations of operating procedures (Schwitter & Fuchs, 1996; Fuchs, Schwertel, & Torge, 1999), and knowledge-based machine translation (Mitamura & Nyberg, 1995).

While controlled languages attempt to reduce ambiguous interpretation of sentences, these benefits are not gained without a cost. Restrictions on the grammar and the lexicon will also limit the expressiveness of the language. The goal is therefore to make the controlled language as expressive as possible while minimizing ambiguity. Consequently, documents in 'standard' English need to be rewritten using the grammar and lexicon of the controlled language, a process that is facilitated by a language that allows a variety of syntactic realizations for the same underlying semantic construct. In addition, a more expressive language also makes the documents more readable for humans.

Although using a restricted grammar and lexicon can reduce the number of interpretations for a sentence, ambiguities often cannot be completely avoided. The

semantics of Qualitative Process Theory (Forbus, 1984) are used to help the interpretation of the sentence. The QP-specific patterns that we introduced in chapter 3 are encoded as grammar rules to capture the underlying QP semantics found in sentences whose syntactic structure can be aligned with those patterns.
The following sections show how a controlled language can be used for describing physical phenomena by balancing the trade-off between limited ambiguity and expressiveness. The language itself is not a formal language but a pragmatically oriented construct that allows enough flexibility to be extended with additional syntactic constructs if needed.

5.1 Types of controlled languages

In the design of a controlled language several questions need to be answered about which parts of the language should be 'controlled' and by which means these restrictions on the language are enforced. There are a number of differing views on what actually constitutes a controlled language, ranging from a very loose degree of 'control' to quite strict and narrow definitions. Restrictions usually appear in two different forms: (a) as restrictions on the grammar, i.e. by controlling the possible syntactic constructs that can be used for sentences, and (b) as lexical restrictions, i.e. by controlling the possible meanings and features each word in the lexicon can take. The combination of these two types of restrictions determines not only the amount of ambiguity of a language but also the habitability of the language (W. C. Ogden & Bernick, 1997; Watt, 1968). A technical writer might create a specialized lexicon as a stripped down version of a regular English dictionary, e.g. one that basically has far fewer entries and only one particular meaning per word. For a computer system, the semantic information included in this lexicon must be created from scratch or derived from some source.

Lexical restrictions are usually implemented by reducing the possible meanings and features for each word in the lexicon. Limiting the number of possible meanings is the most common form of lexical restriction, found in some form in every controlled language. There are three basic types of lexical restrictions to limit the number of meanings of a lexicon entry:

1. Each word in the lexicon has *exactly one part of speech and exactly one possible meaning* assigned to it. Since every available word has exactly one meaning, there is no semantic ambiguity arising from different word meanings. This constraint on the language was first proposed by Ogden in Basic English (C. K. Ogden, 1933, 1937) and has been the guiding principle for the development of early controlled languages. For example, the word 'water' can only be used as a noun and its meaning is always 'the liquid form of H_2O'. However, single word senses are problematic for verbs and prepositions. Although this solution does not require a

separate semantic interpretation process other than just retrieving the semantic information attached to the lexicon entry, it is also very restrictive and inflexible. Because of these reasons, Basic English and its derivatives with larger vocabularies aren't widely used as controlled languages.

2. A more flexible approach is to allow each word to appear in different parts of speech, with *exactly one meaning for each part of speech*. The lexicon could contain two entries for 'water', as a noun and as a verb. Internally, the noun entry might be labeled as WATER1 and the verb entry as WATER2. In a sentence like 'The workers water the tree with water.' the parser would have to determine that the first occurrence of 'water' uses the verb entry WATER2 while the second one uses the noun entry WATER1. Once the parser has identified the parts of speech, the semantic interpretation of the entries is again a simple lookup of the semantic information for each entry. This approach is used by languages such as AECMA (AECMA, 1995) and Simplified English (Verduijn, 2002). It is more flexible than the tight lexical control exerted in Basic English, but it requires a parser or a part of speech tagger to determine the part of speech.

3. Finally, each word could carry *multiple possible meanings for a single part of speech*. Although this approach does not appear to be any different from an unrestricted language, the entries for a word could be reduced to subset of all possible senses found in a dictionary. For example, the OED (Simpson & Weiner, 1989) lists more than a dozen different meanings for the word 'bridge' in the noun sense. Of these meanings, only the two or three most prominent ones might be selected for the lexicon. Again, each entry in the lexicon will have a distinct label and different semantic information attached to it. While this solution provides the greatest amount of flexibility for the user of the controlled language, it also requires a semantic interpretation process that can distinguish the entries from each other and select the one which the most appropriate one. This semantic interpretation step is a non-trivial process and increases in its complexity with each additional ambiguous lexicon entry. Because of this reason, the number of entries with multiple word senses are usually limited, as in Caterpillar Technical English (Nyberg, Mitamura, & Carbonell, 1997). We adopted a similar approach in our system.

Grammatical restrictions are usually implemented by reducing the number of grammar rules so that they only cover particular sentence structures. Reducing or eliminating the ambiguous attachment of complements in noun and verb phrases is the most common grammatical restriction found in controlled languages. For example, syntactic ambiguity arising from prepositional phrase attachment can be controlled by grammar rules that allow only one particular form of attachment directly to the verb instead of various possibilities of nested attachments. The prepositional phrases for the sentence

'The heat flows from the hot brick to the cool ground.' can be attached to the verb in at least two different ways.

(1a) The heat [flows [from the hot brick] [to the cool ground]].
(1b) The heat [flows [from the hot brick [to the cool ground]]].

If the attachment in (1a) is the preferred version, then the grammar of the controlled language should not allow nested prepositional phrases as in (1b), and vice versa.

Another common grammatical restriction is the use of subcategorization information for verbs. The grammar can limit the application of the attachment rules to those verbs that support particular complement structures by using the subcategorization specified in the lexicon entry of the verb. For example, an entry for verb 'move' should allow subcategorizations for its intransitive forms as well as combination of prepositional and noun phrase complements. The sentences (2a) and (2b) would be accepted by the grammar. Intransitive verbs such as 'arrive' cannot take a direct object and do therefore allow no noun phrase complement as subcategorization information in (2c).

(2a) The car moves from the street into the driveway.
(2b) The car moves into the driveway.
(2c) The car arrives.

Coordinated structures are another source of syntactic ambiguity. The following two examples illustrate how conjunctions can be interpreted in at least two different ways in noun and verb phrases.

(3a) [The fluid is warming] and [expanding].
(3b) The fluid [is warming and expanding].
(4a) The fluid consists [of water and oil].
(4b) The fluid consists [of water] and [oil].

Furthermore, syntactic structures can be assigned a particular semantic interpretation, as done in Semantic Grammars (Burton, 1976a). Controlled language grammars usually favor short, clear sentences instead of long and nested constructs. Short sentences with an easy grammatical structure are also easier to understand for human readers and easier to analyze for a parser.

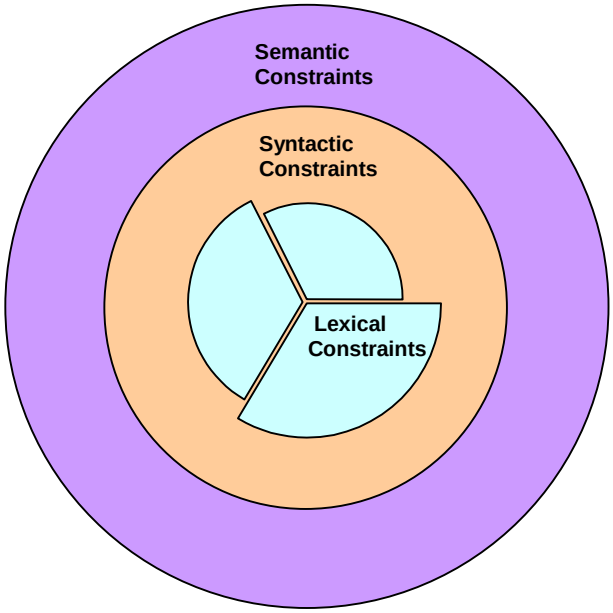

Figure 5.1: Layers of constraints in controlled languages

Figure 5.1 shows the different layers of constraints in controlled languages. At the center of each controlled language are the three different types of restrictions on the lexicon, i.e. the tightly controlled one-to-one mapping, a version that allows different speech parts for a word, and the least restrictive variant that allows multiple ambiguous entries for each speech part but might have a limitations on the overall number of words in the lexicon. One of these three types of restrictions is used every controlled language. Outside the lexical constraints are syntactic constraints imposed on the grammar of the controlled language that define which syntactic structures are acceptable and how phrases attach to each other. On top of these syntactic restrictions, another layer for semantic restrictions can be used to determine the interpretation of particular sentence structures. Certain syntactic structures might be used only in a predefined context like the QP-specific patterns described in chapter 3.

5.2 The Parser

The parser using in our system is a modified version of the publicly available parser described in (Allen, 1995), which is a limited variant of the TRAINS parser (Allen et al., 1995). Our extensions mainly affect in way in which the parser retrieves and processes semantic information.

The parser uses a bottom-up parsing algorithm that constructs an interpretation of a sentence in a compositional manner, starting from terminal nodes. It uses a best-first parsing technique that tries to maximize the length of phrases and sentence structures it can handle. The best parses are generally the ones that handle the longest sequence of words in the input. If a sentence cannot be covered completely, the parser returns a set of phrase and terminal nodes that it could handle. This partial parse data is still valuable for the semantic interpretation process.

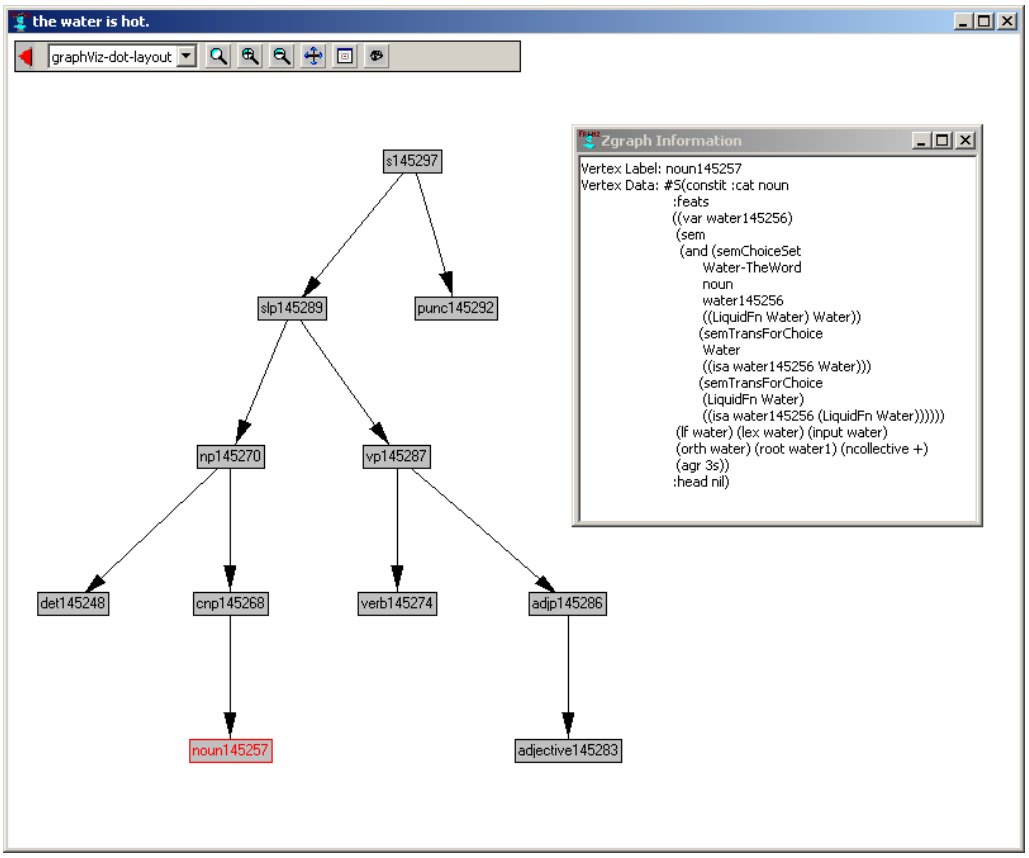

Figure 5.2: Parse tree for 'The water is hot.'

Figure 5.2 shows a complete parse tree for the sentence 'The water is hot.' The leaf node nodes of the tree are the terminal nodes corresponding to the individual of the words of the sentence. The parser uses grammar rules to combine information from these nodes to form phrase nodes. For example, the determiner 'the' and the common noun 'water' can be combined into a common noun phrase (CNP), which in return can be used to form a sentence-level phrase (SLP) together with a verb phrase. We will discuss the grammar rules and the support for QP-specific structures shortly.

The parser makes extensive use of a feature system, including feature percolation. Figure 5.2 shows the feature information associated with the expanded terminal node for the noun ‘water’. Using feature percolation, information such as the semantic data for a node is ‘moved up’ to the phrase head when new phrases are constructed from constituent nodes. Using this technique, information such as variable names and semantic data attached to terminal nodes can be passed on to phrase nodes. We do not compute logical forms as described in (Allen, 1995), but rely on the semantic information provided by the Cyc knowledge base contents. Chapter 6 addresses the interaction between the parser and the knowledge base for the retrieval of general semantic data in detail.

5.2.1 The Lexicon

The lexicon for the syntactic parser is a feature-based collection of lexical entries derived from the COMLEX 3.1 lexical database (Macleod, Grishman, & Meyers, 1998), which we acquired from the LDC consortium. The COMLEX data, consisting of approximately 38,000 entries, was transformed into a format that can be used with the parser. In addition to a number of new features required by the parser, the original feature information of the COMLEX data was preserved. The additional features include the possible agreement information for noun and verb entries as well as markers for comparative and superlative form for gradable adjectives and adverbs. The lexicon uses a format that organizes each entry by its lexical form, a set of features, and a unique identifier. Figure 5.3 shows an example lexicon entry.

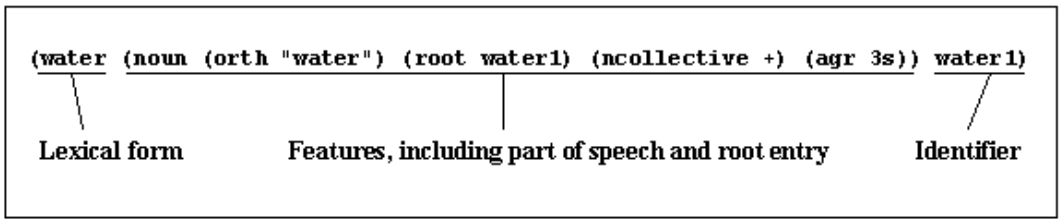

Figure 5.3: Lexicon entry format

The parser does not provide a morphology component, so we had to expand the COMLEX data to include all possible morphological variants of each entry. We included all plural forms for nouns, every inflected form including the base form for verbs, and the comparative and superlative forms for adjectives and adverbs. The size of the expanded lexicon effectively doubled in size and consists of more than 86,000 entries in its current version.

Expanding the lexicon instead of using a morphological analyzer means trading off extra memory space occupied by the additional lexicon entries against the processing time spent on a morphological analysis. For example, one version of the Allen's TRAINS system that used the Alvey morphological analyzer (Ritchie, Russell, Black, & Pulman, 1991) spent 20 minutes on parsing the word 'reconsideration' using a Sparc 10 workstation, as reported in (Allen et al., 1995). Since modern hardware provides at least an order of magnitude more memory (and processor power), using an expanded lexicon, as implemented in our system, is an alternative to an inline morphological analysis.

To avoid loading the entire set of entries into memory, the lexicon data itself is stored in the background knowledge base. The parser itself only maintains a small temporary lexicon that contains all entries for words encountered since startup. Entries are retrieved on demand from the KB, cached in the temporary lexicon to reduce retrieval times, and then made available to the parser.

The following examples for the noun 'temperature', the verb 'heat', and the adjective 'hot' illustrate how COMLEX entries are transformed into a format that fits the parser.

5.2.1.1 Example 1: Temperature

The COMLEX entry for the noun 'temperature' specifies only two features, indicating that the entry is a count noun and refers to a scale. The semantic information about scales ('dimensions', 'quantity types') included in the lexicon is not used in the semantic interpretation process. Instead, we use the background knowledge base to retrieve information about quantity types and units. The goal was to separate semantic knowledge from lexical knowledge to make sources such as the lexicon more maintainable.

COMLEX entry:

```
(noun :orth "temperature"
      :features ((nscale)(countable)))
```

The modified entries include all the information found in the original COMLEX entry and add information about agreement and the root lexical entry. The root is usually a numbered identifier corresponding to the base entry form. In this case, the singular and plural form of the noun *temperature* share the same root identifier, *temperature1*. The root identifier links plural entry of the word to the singular form.

Modified lexicon entries:

```
(temperature
 (noun (orth “temperature”) (root temperature1)
       (nscale +) (countable +) (agr 3s))
 temperature1)

(temperatures
 (noun (orth “temperatures”) (root temperature1)
       (nscale +) (countable +) (agr 3p))
 temperatures1)
```

5.2.1.2 Example 2: Heat

The verb entry for *heat* in the COMLEX lexicon includes a number of different subcategorization features. Each of these features is transformed into corresponding entries for the **subcat** feature. The particle information in the **pval** and **adval** features is not used by current version of the parser, but it can be added to future versions of the lexicon.[1]

COMLEX entry:

```
(verb :orth “heat”
      :subc ((np-pp :pval (“to”))
             (part-pp :adval (“up”)
                      :pval (“into” “on” “over” “to”))
             (part-np :adval (“up”))
             (part :adval (“up”))
             (part-np-pp :adval (“up”)
                         :pval (“with”))
             (np)
             (intrans)))
```

Lacking a morphology component in the parser, the single entry for the verb *heat* has to be expanded into six entries in the modified lexicon to cover all possible inflected forms of the verb. Note that each entry makes use of the verb form (**vform**) and agreement (**agr**) features to indicate the particular usage of the inflected verb entry.

Lexicon entries for different parts of speech can have the same basic orthographic form. For example, the noun *heat* has the same basic form as the verb. The entry for the noun is listed in our lexicon as *heat1*, so the base form of the verb starts out as *heat2*. Similarly, the present tense form generated from the verb entry has the identifier *heat3* to distinguish it from the base form. However, the root feature information for

---

[1] The **adval** feature specifies adverbial particles that can be used with a subcategorization of a verb. The **pval** features specifies prepositions that can be used as the head of a PP complement phrase in a subcategorization.

heat3 will tie it to the base verb entry.[2] This is a lexical differentiation between different parts of speech and not some form of word sense disambiguation. The noun *heat* itself can have multiple different word sense, yet syntactically they are all collapsed into a single lexicon entry, *heat1*. On the other hand, the two verb entries, *heat2* and *heat3* originate both from the same verb sense.

Modified lexicon entries:

```
(heat
 (verb (orth “heat”) (root heat2) (vform base)
       (subcat (? s np-pp part-pp part-np part
                  part-np-pp np intrans)))
 heat2)

(heats
 (verb (orth “heats”) (root heat2)
       (vform pres) (agr 3s)
       (subcat (? s np-pp part-pp part-np part
                  part-np-pp np intrans)))
 heats2)

(heated
 (verb (orth “heated”) (root heat2)
       (vform past) (agr (? a 1s 2s 3s 1p 2p 3p))
       (subcat (? s np-pp part-pp part-np part
                  part-np-pp np intrans)))
 heated2)

(heated
 (verb (orth “heated”) (root heat2) (vform pastpart)
       (subcat (? s np-pp part-pp part-np part
                  part-np-pp np intrans)))
 heated1)

(heating
 (verb (orth “heating”) (root heat2) (vform prespart)
       (subcat (? s np-pp part-pp part-np part
                  part-np-pp np intrans)))
 heating1)

(heat
 (verb (orth “heat”) (root heat2) (vform pres)
       (agr (? a 1s 2s 1p 2p 3p))
       (subcat (? S np-pp part-pp part-np part
                  part-np-pp np intrans)))
 heat3)
```

[2] The present tense entry contains agreement information, while the base form does not.

5.2.1.3 Example 3: Hot

The entry for the adjective *hot* in the COMLEX states that the adjective is gradable. In this case, entries for the comparative and superlative forms have to be added to our expanded lexicon.

COMLEX entry:

```
(adjective :orth “hot”
           :features ((gradable :both t)))
```

The expanded lexicon contains three entries for the adjective *hot*, each labeled as gradable and with feature information for their corresponding degree, i.e. comparative and superlative. As in the previous examples, all three entries share the same root feature information to indicate their common heritage. Note that the entries do not contain any semantic information such that *hot* is conceptually related to the noun *temperature*. As stated above, the semantic interpretation process will rely on the background knowledge base to retrieve such information.

Modified lexicon entries:

```
(hot
 (adjective (orth “hot”) (root hot1) (gradable +))
 hot1)

(hotter
 (adjective (comparative +) (orth “hotter”)
            (root hot1) (gradable +))
 hotter1)

(hottest
 (adjective (superlative +) (orth “hottest”)
            (root hot1) (gradable +))
 hottest1)
```

The lexicon allows multiple entries of the same word for different parts of speech as well as multiple word senses for each entry, and consequently, lexical and semantic ambiguities have to be resolved. The parser resolves lexical ambiguity by committing to a particular part of speech when it produces a parse tree. Although all possible parts of speech are considered during the parse, only one, that fits best in the longest sequences of words covered, is selected. Ambiguous word senses are resolved during the semantic interpretation process by a word sense disambiguation module discussed in chapter 6. The following section describes how the QP-specific patterns discussed in chapter 3 can be integrated with general syntactic constructs and implemented as a grammar for the parser.

5.3 Describing physical processes in natural language

We have developed QRG-Controlled English (QRG-CE, in short) as an input language for describing physical processes using natural language. Three major design decisions were made in the development of QRG-CE that shaped the language itself and the way in which it is used. First, the language has to support the natural language patterns that express the constituents of Qualitative Process Theory in natural language, as discussed in Chapter 3. Second, we decided that expressiveness is an important factor for the habitability of the language. Therefore, the controlled language allows multiple word senses for each part of speech. In the system we have implemented, we use a semantic interpreter to resolve remaining ambiguities. Finally, QRG-CE has to be flexible enough to allow extensions, i.e. the grammar has to be open. Similar to the way in which human natural languages change over time, we will allow QRG-CE to change and adjust to its uses. For example, we have analyzed our corpus for common syntactic patterns in which constituents of QP Theory are expressed, but it cannot be assumed that this list of pattern is final. Additions to the current corpus might reveal new syntactic patterns for expressing QP-relevant information. It is expected (and desired) that QRG-CE will change with continued use, and we have to account for this fact. The language has been implemented as a grammar for the bottom-up chart parser described in the previous section.

The current version of QRG-CE is intended for writing descriptions of single instances of physical processes and leaves out more complex types of descriptions. For example, in descriptions that deal with behaviors over time the parser and interpreter would need to handle temporal information, i.e. the tense and aspect of sentences, and construct temporal representations. General descriptions of physical processes are also not supported. Besides the fact that quantification is necessary to handle general process descriptions, our corpus analysis has shown that general and instance information is often mixed within the same description. This would require a mechanism to distinguish general facts from knowledge specific to an instance. These issues will be discussed in more detail as future work in chapter8. The following section provides an overview of the syntactic structures that the grammar can handle and those that still present challenges.

5.3.1 Support of the constituents of QP Theory

QRG-CE covers common syntactic structures found in standard English that provide the foundation of the controlled language. In addition, it includes support for the syntactic patterns that are used to express QP-related information.

The key component of QRG-CE is a set of grammar rules that provide syntactic constructs to capture information related to QP Theory. Direct and indirect influences

as well as ordinal relations appear in several different forms in unrestricted natural language text. These forms can be grouped into larger classes of patterns and each of these patterns can be captured as grammar rules for QRG-CE. Chapter 3 contains a detailed overview of the individual patterns illustrated by examples sentence from the corpus. Coupling syntactic patterns with a particular semantic interpretation traces back to the ideas of Semantic Grammar (Burton, 1976a) and is used in many dialog systems and natural language interfaces to databases, e.g. (Allen et al., 1995; Androutsopoulos, Ritchie, & Thanisch, 1995; Copestake & Sparck Jones, 1990).

The following section illustrates how the QP-specific patterns can be encoded as grammatical rules. The parser uses a context-free grammar that is similar to the grammar structure described in (Allen, 1998). Examples for a few simple rules for a noun and verb phrases are shown in Figure 4.

For example, the following rule for the construction of a common noun phrase (CNP) consists of three parts – the left-hand side parent `(cnp (agr ?a) (var ?varn) (sem ?semn))`, a name `–cnp->n-` , and a right-hand side constituent `(head (noun (agr ?a) (var ?varn) (sem ?semn)))`.[3]

```
((cnp (agr ?a) (var ?varn) (sem ?semn))
  -cnp->n-
  (head (noun (agr ?a) (var ?varn) (sem ?semn))))
```

The parent defines the type of (phrase) node and its features that will be constructed from one or more right-hand side constituents. Variables bound for features on the right-hand side will be unified with the parent constituent. In the example above, the values of the features for agreement (**agr**), variable (**var**), and semantic information (**sem**) will be simply passed from the noun on the parent common noun phrase.

[3] The grammar rules also allow a (numeric) weight to be specified between the name and the right-hand side constituents. Weights are used by the parser for computing the best interpretation. The best-first search strategy of the parser prefers rules with higher weight when constructing phrase nodes. The default value for each rule is 1.

Not every feature has to be explicitly specified on the left- and right-hand side of the rules. Head features can be defined to enforce feature constraints between the mother constituent (i.e. the left-hand side of the rule) and the child subconstituents marked as heads (on the right-hand side of the rule). Features declared as head features for a particular phrase type always have to match between those constituents. For example, the head constituents of verb phrases typically use the **vform** feature. Even though the last rule in figure 5.4 does not list this it, the **vform** feature of the parent and the right-hand side constituents have to be identical.

A Common Noun Phrase:

```
((cnp (agr ?a) (var ?varn) (sem ?semn))
 -cnp->n-
 (head (noun (agr ?a) (var ?varn) (sem ?semn))))
```

A Noun Phrase consisting of a CNP with a determiner:

```
((np (var ?varcnp) (agr ?a) (sem ?semcnp))
 -np->det-cnp-
 (det (agr ?a))
 (head (cnp (agr ?a) (var ?varcnp) (sem ?semcnp))))
```

A Simple Verb Phrase:

```
((svp (var ?varv) (sem ?semv) (agr ?agr) (subcat ?subc))
 -svp->v-
 (head (verb (var ?varv) (sem ?semv) (agr ?agr) (subcat ?subc)))
```

A Verb Phrase consisting of a subcategorized simple verb phrase and its NP complement:

```
((vp (var ?varv) (agr ?agr) (:OBJECT ?varnp)
     (sem (and  ?semv ?semnp)))
 -vp->svp-np-
 (head (svp (subcat np) (agr ?agr) (var ?varv) (sem ?semv)))
 (np (var ?varnp) (sem ?semnp)))
```

Figure 5.4: Grammar rules

5.3.1.1 Encoding QP-specific patterns as grammar rules

The QP-specific patterns discussed in chapter 3 can be encoded as grammatical rules. For example, one of the patterns for indirect influences used in a sentence like “As the volume of the air increases, the density [of the air] decreases.” Is:

```
AS <Quantity1> <Change1>, <Quantity2> <Change2>.
```

This pattern is covered by the following rule for sentence-level pattern (SLP). Sentence-level patterns include general clause structures as well as the QP-specific patterns.

```
((slp (var ?vars)
    (sem (and ?sems1 ?sems2
              (qpropEvent ?vars1 ?vars2))))
 -slp->as-slp-comma-slp-
 (sconj (lex as))
 (head (slp (var ?vars1) (sem ?sems1)))
 (punc (lex punc-comma))
 (head (slp (var ?vars2) (sem ?sems2))))
```

The rule does not explicitly require the use of two quantities and two change events, and will work for any two sentences used as subordinate constituents. This is a feature of the layered architecture of our system to separate the semantic interpretation process from the syntactic parsing step. It is therefore the task of the semantic interpreter to verify that the information from the two subconstituents meets the semantic requirements. A sentence like ‘As the lights go out, the audience begins to cheer.’ can be parsed, but it will not be interpreted semantically as an indirect influence pattern.

Similarly, patterns for causal links between events can be encoded as grammar rules. For example, patterns using subordinating conjunctions such as ‘if’ or ‘because’ express causal relationships between two parts of a sentence. The following example shows how a causal relationship via the conjunction ‘because’ can be captured in grammatical rules.

Pattern:

<Event1>, BECAUSE <Event2>

Example:

“The heat flows from the brick to the ground,
because the brick has a greater temperature than the ground.”

Grammar rule:

```
((slp (var ?vars)
   (sem (and ?sems1 ?sems2
                  (causesEventEvent ?vars2 ?vars1))))
 -slp->slp-comma-because-slp-
 (head (slp (var ?vars1) (sem ?sems1)))
 (punc (lex punc-comma))
 (sconj (lex because))
 (head (slp (var ?vars2) (sem ?sems2))))
```

Ordinal relations in sentences such as ‘The temperature of the brick is greater than the temperature of the ground’ can also be captured by grammatical rules. The main difference from the rules presented so far is that that QRG-CE uses phrase-level rules instead of sentence-level rules for capturing information about ordinal relations. The semantic information for the main rule for comparison phrases uses keywords to capture the compared elements and the relationship between them. The actual comparison is represented as an event with reified role relations containing keywords that will be replaced with the semantic information from other phrases when the comparison phrase data is passed upwards to its parent nodes. This allows the parser to incrementally accumulate information about a comparison expressed in a sentence.

```
((compp (var ?varnp)
        (sem (and ?semnp
                  (isa ACTION ComparisonEvent)
                  (comparer :ACTION :COMPARED1)
                  (comparee :ACTION :COMPARED2)
                  (compareReln :ACTION :COMPREL))))
 -compp->than-np-
 (cconj (lex than))
 (head (np (var ?varnp) (sem ?semnp))))
```

The following adjective phrase rule illustrates how a comparison phrase can be used as a subconstituent in sentences such as ‘The car is [faster than the truck].’ When the rule is applied, the keywords `:COMPREL`, `:COMPARED1`, and `:COMPARED2` are substituted with other keywords or replaced by values bound to variables in the parent constituent (i.e. the left hand side of the rule).

```
((adjp (var ?varadj) (sem ?semcompp) (:COMPREL ?semadj)
       (:COMPARED1 :SUBJECT) (:COMPARED2 ?varcompp))
 -adjp->adj-compp-
(head (adjective (comparative +) (var ?varadj) (sem ?semadj)))
(compp (var ?varcompp) (sem ?semcompp)))
```

5.3.1.2 QP-specific content not encoded in the grammar

Direct influences are generally indicated by a verb of the transfer, motion, change or flow domain. In most cases, the sentence also uses prepositional phrases to indicate the source and destination of the transfer.

Pattern:

<Quantity> <Change> [from <location1>]
[to <location2>] [<path>]

Example:

“The heat flows from the stove to the kettle.”

The grammar for QRG-CE does not contain rules for capturing information about direct influences. Direct influences are usually connected to a particular semantic class of verbs and each particular use of these verbs resulting in direct influences would have to be encoded as a separate grammar rule, including a number of checks for semantic constraints. This approach results in time-consuming verification of semantic restrictions. The detection of direct influences is therefore deferred to the semantic interpretation process that follows the syntactic parse. The semantic interpreter uses information from the background knowledge base to identify event structures associated with particular verbs and will instantiate the appropriate DirectInfluence frames. Chapter 6 discusses the interpretation of direct influences and other QP constituents in more detail.

5.3.2 General syntactic constructs

While the grammar for QRG-CE provides basic syntactic constructs of standard English, this section addresses a number of issues that are important for understanding the syntactic constraints on QRG-CE.

QRG-CE uses grammatical constraints on two different levels. The first set of constraints affects the construction of individual phrases such as noun phrases, verb phrases and prepositional phrases from lexemes, e.g. the construction of a verb phrase from a verb and a particle. It also includes the construction of phrases from lexemes and phrases, such as the construction of a noun phrase from a determiner and a common noun phrase. The combination of phrases is also covered by this level, e.g. the construction of a verb phrase from a verb phrase and its complement noun and prepositional phrases.

Phrase level:
(a) combination of lexemes into phrases,
(b) the combination of lexemes and phrases,
(c) and the combination of phrases

Sentence level:
(a) combination of sentential structures,
(b) the combination of phrases into sentential structures

Figure 5.5: Grammatical constraints in QRG-CE

The second set of constraints determines how combinations of phrases are handled on the sentence level, and how complete sentence structures are formed from phrases. This level also includes constraints that affect how sentential structures themselves can be combined by using coordinate and subjunctive conjunctions. Figure 5.5 shows an overview of the two levels of grammatical constraints in QRG-CE.

The grammatical constraints discussed in this section are general restrictions to limit ambiguity. Similar constraints can be found in other controlled languages.

5.3.2.1 Phrase-level grammatical constraints

The phrase-level of the QRG-CE grammar determines how lexemes and phrases can be combined into higher-order phrases. While the grammatical rules allow many syntactic constructs of standard English, there are a number of important restrictions that the user of the controlled language need to take into account.

- *Definition of proper nouns.* Proper nouns have to be defined in the lexicon as named entries to be correctly recognized. A proper noun phrase (PNP) will only be created if at least one of the constituents is defined as a proper noun. Proper nouns can be used together with a determiner, as in 'A Volkswagen is of better quality than a Yugo.' Proper nouns cannot be defined 'on-the-fly' but must be added as to the lexicon up front to be recognized by the language.

- *Unknown entries as discourse variables.* Undefined words are flagged as an Unknown entry, and can only be used as names for variables together with a common or proper noun. Unknowns cannot build a noun phrase with an accompanying noun. For example, 'the can C1' is a legal noun phrase, while 'C1' or 'the C1' are not. The Unknown will be used as the discourse variable for the noun it is associated with.[4] This is the only way to specify consistent discourse variables for entities across different sentences in a paragraph. Even if the unknown has been introduced together with an object before, the grammar requires the using a common or proper noun with each subsequent mentioning of the variable. For example, in the sentences 'The pipe P1 connects the cylinder C1 and the cylinder C2. The cylinder C1 is filled with water.' the word 'cylinder' is required in the second sentence. The parser does not check semantic restrictions imposed by the discourse variables, i.e. that the entities associated with the variable C1 are both cylinders. If the second sentence would read 'The can C1 is filled with water.' the parser would accept it as well. The semantic interpreter would have to ensure that two entities are identical before any of the information

[4] In most other cases, the parser will use the variable associated with the head constituent of the phrase.

associated with them could be merged. It is therefore recommended to use unknowns as discourse variables only with the exact same noun to ensure consistency in the semantic interpretation process. Although this is a quite conservative merge strategy, it avoids potential problems arising from inconsistent concept definitions in the background knowledge base.

- *Verb particles.* The use of verbs with particles such as ‘start up’, ‘decide on’ or ‘get over’ is problematic, because the particle is easily confused with a preposition and might generate undesired interpretation. ‘John is deciding on a car.’ means that John is about to select a car, not that he is sitting or standing on a vehicle to make a decision. It is recommended to entirely avoid using verbs with particles. Verb particles are currently interpreted as prepositions. For example, the phrases ‘deciding on a car’ and ‘sitting on a car’ will both be interpreted as a verb with a prepositional phrase complement. Alternatives are readily available: ‘start’ instead of ‘start up’, ‘select’ instead of ‘decides on’, or ‘forget’ instead of ‘get over’. If in doubt, the user might consult a resource such as WordNet to find appropriate synonyms.

- *Coordination in verb phrases.* The use of coordinated verbs is another major source of ambiguity. For example, ‘The liquid boils and expands in the cylinder.’ can be interpreted in more than one way. Does the liquid just boil, or does it boil in the cylinder? To eliminate this ambiguity, the language does not support coordinated verb phrases. Splitting this sentence into two ‘The liquid boils in the cylinder.’ and ‘The liquid expands in the cylinder.’ resolves the interpretation problem.

- *Conjoined prepositional phrases.* Similar to coordinated verb phrases, the use of conjoined prepositional phrases could lead to ambiguous interpretations and should therefore be avoided. For example, in the sentence ‘The cylinder contains 200 milliliters of water and alcohol.’ it is possible that the cylinder contains a 200 ml mixture of water and alcohol in unspecified proportions, or it contains 400 ml of fluids, i.e. 200 ml of water and 200 ml alcohol. Rewriting the sentence as ‘The cylinder contains 200 milliliters of water and 200 milliliters of alcohol.’ resolves the ambiguity. Note that this kind of coordination appears to occur only for certain prepositions such as ‘of’, but not for others such as ‘from’ or ‘to’ that cannot take coordinated noun phrases.

- *Units and values.* Measure phrases require a value (either as a number or written out) and a unit in its non-abbreviated form. A value without a unit or a unit without an accompanying value are not valid measure phrases and should be avoided. For example, the phrases ’ten liters’ and ‘500 kilometers’ will be accepted, while ‘ten’ or ‘kilometers’ by themselves are not recognized as valid measure phrases by the

grammar. Furthermore, an important lexical restriction applies to measure phrases. Lexical items denoting units have to have a one-to-one mapping and cannot be used in any other word sense for the same part of speech. This restriction helps the semantic interpretation process and hardly limits the expressiveness of the controlled language. For example, the word 'meter' can still be used in the verb sense and the unit 'Fahrenheit' can still be used as proper noun, since measure phrases do not take verbs or proper nouns as any of their constituents.[5]

5.3.2.2 Sentence-level grammatical constraints

Sentence-level grammatical constraints are restrictions that involved the construction a sentence structure from complete phrases.

- *Coordinate conjunctions.* Coordinate conjunctions such as 'and' or 'or' can only be used to join complete sentence structures. QRG-CE also requires the use of a comma before the coordinate conjunction. For example, the sentence 'He walks to the house' cannot be extended by the conjunction 'or' and the verb (phrase) 'runs' to form the sentence 'He walks to the house or runs.' The second part should be a complete sentence such as 'he runs', which would result in the acceptable sentence 'He walks to the house, or he runs.' Even more preferable for a successful interpretations is a more elaborate and balanced form that repeats the object of the first part, if that is the meaning intended by the author: 'He walks to the house, or he runs to the house'.[6]

- *Subordinating conjunctions.* Sentences with subordinating conjunctions expect a complete sentence structure to follow the conjunction. A comma is required before the conjunction or after the sentence structure, depending on the overall form of the sentence. For example, 'The water is flowing out of the can, while the valve of the cylinder remains closed.' is an acceptable sentence.

- *Punctuation.* Punctuation is required to help the parser with the identification of complete sentence structures. A sentence has to end with a period, a question mark, or an exclamation mark. Within a sentence, the comma, the colon, and the semicolon also separate subordinate clauses.

---

[5] A more problematic case would be the word 'bar', which can denote a unit of atmospheric pressure as well as a drinking establishment. Depending on how important the latter word sense is for the input text, a possible solution would be to use an alternative physical unit (such as Pascal) in measure phrases.

[6] As it had been discussed earlier, a coordination of the verbs ('He walks or runs to the store') would not be a good solution either.

5.3.3 Limitations of QRG-CE

QRG-CE allows a large variety of general syntactic structures and uses a number of restrictions, as discussed in the previous section. Nevertheless, there are a number of challenges for the QRG-CE and for controlled languages in general. It is important to note that the issues in this section concern syntactic and semantic limitations, such as the attachment of phrases or particular clause structures. Limitations of the semantic interpretation process will be discussed in more details within the context of the semantic interpreter in chapter 6.

- *Compound nouns.* While the grammar could easily handle this type of nouns, they are problematic for the semantic interpretation process. Leaving out compound nouns is not a technical issue or an artificial limitation of the grammar or the parser. It reflects the difficulties humans have with interpreting compound nouns in general (Wisniewski & Gentner, 1991; Wisniewski & Murphy, 1989). For example, a ‘master key’ is essentially a key, while an ‘ice cube’ is primarily a chunk of ice that happens to have the shape of a cube. To avoid these interpretation issues, the relationship between the two words can be made explicit (‘level of water’ instead of ‘water level’) or the words should form a hyphenated new lexical entry (‘ice-cube’ instead of ‘ice cube’).

- *Single subjects and coordinated noun phrases.* The subject of a sentence has to be single common or proper noun as the head of a noun phrase. The noun phrase can contain a determiner and an unknown that is used as the discourse variable for the noun. For example, ‘house’, ‘the house’, and ‘the house H1’ are all acceptable subjects for a sentence. Coordinated noun phrases such as ‘the can and the cylinder’ should be avoided in general. For example, the sentences ‘The can and the cylinder are connected by the pipe.’ can be rewritten as ‘The pipe connects the can to the cylinder.’ The sentence ‘The cylinder contains water and oil.’ can be rewritten as two sentences. The main problem with coordinated noun phrases is the manner in which discourse variables are currently treated in the QRG-CE grammar. The discourse variable for a noun phrase is tied to the head noun of the phrase. In coordinated noun phrases two or more nouns become constituents of a superordinate noun phrase that would generate a new discourse variable. A possible solution to this problem would be the use of specially marked ‘collective discourse variables’ in superordinate noun phrases that contain the actual discourse variables for the head nouns. Once the parse is complete, all semantic information that uses the collective variable would need to be replicated for each original variable.

- *Variable attachment of prepositional phrases.* Prepositional phrases are generally attached to the head of the current noun or verb phrase, i.e. nearest preceding main verb or noun (left attachment). In sentence (5) both prepositional phrases, 'from the brick' and 'to the ground' are attached to the verb 'flow'.

  (5) The water flows from the brick to the ground.

- *Leading or trailing adverbial phrases.* While phrases such as 'So,…' or 'Moreover, …' make a text more readable and are important elements of discourse processing. The grammar rules allow these phrases, but the information associated with them (if any) is not used in the construction of the semantic information for the rest of the sentence. Furthermore, the grammar requires the leading or trailing adverbial phrase to be separated by a comma from the remainder, which must be a complete sentence structure.

- *Passive voice.* Sentences using passive voice are handled by the QRG-CE grammar in a limited way. However, the semantic interpreter process does not distinguish between active and passive voice. The information retrieved from the knowledge base does not include voicing features to allow a differentiation between different argument structures in expressions. For example, voicing information is needed to fill the knowledge base pattern `(contains :SUBJECT :OBJECT)` for sentences (7) and (8). While this pattern is correctly filled for sentence (7), the arguments would be assigned incorrectly for sentence (8).

  (7) The box contains the ball.
  (8) The ball is contained in the box.

  Limiting sentences to the use of active voice is a common restriction found in controlled languages (Almquist & Sagvall Hein, 1996; Fuchs & Schwitter, 1996; Mitamura & Nyberg, 1995).

- *Past and future forms for verbs.* Another common restriction on controlled language is the exclusive use of the present tense. Although the grammar of QRG-CE allows sentences in the past and future tenses of standard English, the semantic interpretation process currently ignores the tensed verbs and assumes present tense. Processing tense information would require the use of representations for expressing temporal relationships such as (Allen, 1984) in the general semantic

interpretation. This goes beyond the scope of the present work and constitutes an area of future work.[7]

- *Relative clauses.* QRG-CE is able to handle simple relative clauses that are co-referential to the subject or the object of the sentence. These relative clauses are introduced by the pronoun 'that or a wh-pronoun. However, to reduce the possibility of a misinterpretation of the sentence, it is preferable to avoid such constructs and rewrite the sentence. Sentences (10) and (11) show a rewritten version of (9) that eliminates the relative clause and the passive voice.

  (9) The liquid, which was poured into the cylinder, flows through the pipe into the tank.
  (10) The cylinder contains the liquid.
  (11) The liquid flows through the pipe into the tank.

  Complex relative clauses are significantly more difficult to handle and have been omitted from QRG-CE. For example, sentences that use pronouns in genitive relation to the head of the noun phrase ('The man whose son was in your class has married again.') or prepositions in conjunctions with a relative pronoun ('The year in which the earthquake happened …', 'The instrument with which the glass was broken has been found.') are not supported.[8]

5.4 Examples

The following section presents two examples to demonstrate how QRG-CE can be used to describe instances of simple physical processes.[9] We will first describe the process in unrestricted natural language, followed by a version that uses QRG-CE and can be processed by our system. The same examples are used again in Chapter 7, when we present full semantic interpretations generated from the input text.

---

[7] The exclusive use of the present tense is a technique used in introductory self-study material for foreign language learning (Hammitt, 1997). Even in everyday conversations, events that happened in the past are often reported in present tense, as in 'This guy comes up to me and asks me for the time …'

[8] This restriction can also be found in other controlled languages (Mitamura & Nyberg, 1995).

[9] Readers who are familiar with the classic QP literature will certainly recognize these examples.

5.4.1 Fluid flow between containers

The unmodified description:
"Two cylinders C1 and C2, connected by a pipe, each contain a certain amount of water. Because the water level in the first cylinder is higher than the water level in the other, water will from cylinder C1 into cylinder C2."

The rewritten text:
(1) A pipe connects cylinder C1 to cylinder C2.
(2) Cylinder C1 contains water.
(3) Cylinder C2 contains water.
(4) Water flows from cylinder C1 to cylinder C2, because the level in C1 is higher than the level in C2.

While in the original version a single sentence establishes the scenario for the physical process description, the rewritten QRG-CE version uses three individual sentences. The information about the connection between the two containers is given in the first sentence. Unlike the two cylinders, the pipe is not named in this example and will be assigned a default discourse variable.

The following two sentences, (2) and (3), are required because of a limitation in the way the semantic interpretation is generated by QRG-CE grammar rules. A more compact, single sentence version such as 'The two cylinders C1 and C2 contain water' has two distinct individuals in the noun phrase and would require a duplication of the semantic information supplied by the verb phrase, as discussed earlier. For this reason, the information has to be explicitly stated for each individual.

Sentence 4 remains almost unchanged from its counterpart in the original text. Although it reverses the clauses, QRG-CE can handle both versions. The sentence contains two QP-specific patterns: the first part uses a pattern for describing a transfer of a quantity, while the second part utilizes a quantity-neutral pattern for a comparison between quantities. Furthermore, the rewritten version does not use compound nouns.

5.4.2 Conduction heat flow – the ice cube on a metal rod

The unmodified description:
"A metal rod with an ice cube on one end is placed in a cup of hot coffee. The ice cube begins to melt, because heat is conducted by the rod from the coffee to the ice cube."

The rewritten text:
(1) An icecube is connected to the end of a metal rod.
(2) The rod is placed in a cup of hot coffee.
(3) The rod conducts heat from the coffee to the icecube.
(4) The heat causes the icecube to melt.

The first two sentences of the rewritten version contain the information of the more compact version in the original text. Note that the compound nouns are handled in two different ways. The 'ice cube' is contracted into a single noun, while the 'metal rod' stays unchanged. 'Metal' is treated as a quality of the rod and becomes an adjective. If this is not the desired interpretation, it could also be contracted into 'metalrod', assuming that an appropriate lexicon entry exists for it.

Splitting the second sentence into two separate sentences helps the semantic interpretation process by explicitly stating a consequence of the heat transfer process.[10]

5.5 Summary

The main goal in the design of QRG-CE was to achieve a high level of expressiveness, without adding ambiguity. The linguistic constructs realized in QRG-CE are intended to primarily cover descriptions of physical phenomena. Textbooks and especially popular science texts usually embed QP-related pieces of knowledge in other information that is not needed for reconstructing the actual underlying process. The use of a controlled language allows us to focus on the relevant parts of the process descriptions, i.e. those that can be processed and interpreted by the semantic interpreter.

The use of a controlled language for the interpretation of physical phenomena brings up an interesting issue – the processing of metaphors with a physical basis. A sentence such as 'The man exploded when the computer crashed.' will be handled appropriately by the QRG-CE parser because it is not syntactically different from a sentence such as 'The gas tank exploded when the car crashed'. The grammar rules that are applicable to descriptions of physical processes can also handle metaphors with a physical basis. However, while the QRG-CE grammar can be used without modifications for this kind of metaphors, the semantic interpretation process needs some adjustments. For

[10] There is one subtle component for the construction of a process model missing in this description. The original text assumes knowledge about an essential property of the ice cube, i.e. that it is cold. The temperature difference between the ice cube and the coffee causes the heat two flow. While the text includes a consequence of heat flow process, it does not mention the condition.

example, humans rarely explode in a physical way by being blown up into bits and pieces. If the noun used with ‘to explode’ is a living thing, the verb should be interpreted as ‘getting very angry.’ Since the semantic interpretation process uses the contents of the knowledge base, new data could be added that provides the additional semantic information and the restrictions on the arguments. We will address the interpretation of metaphors as future work in chapter 8.

Chapter 6

Semantic Interpretation

Almost every natural language processing system that produces representations from textual descriptions uses a syntactic parser and a semantic interpretation process for analyzing their input. Our system follows this modular approach and builds models of physical processes from natural language descriptions in two distinct steps. First, the input is subjected to a syntactic analysis by the parser, which generates a list of parse trees that correspond to all possible syntactic interpretations based on the grammar it uses. The results of the parsing step are the used in semantic interpretation process to produce particular domain-specific representations of the descriptions.

One part of the semantic interpretation process is integrated with the syntactic analysis. The parser itself produces a general semantic interpretation based on the grammar rules for the controlled language and general semantic information retrieved from the background knowledge base. The resulting interpretation does not contain any QP frame structures yet, but it includes supporting information about QP-specific patterns as well as unresolved choice sets for words that map to multiple concepts in the knowledge base.

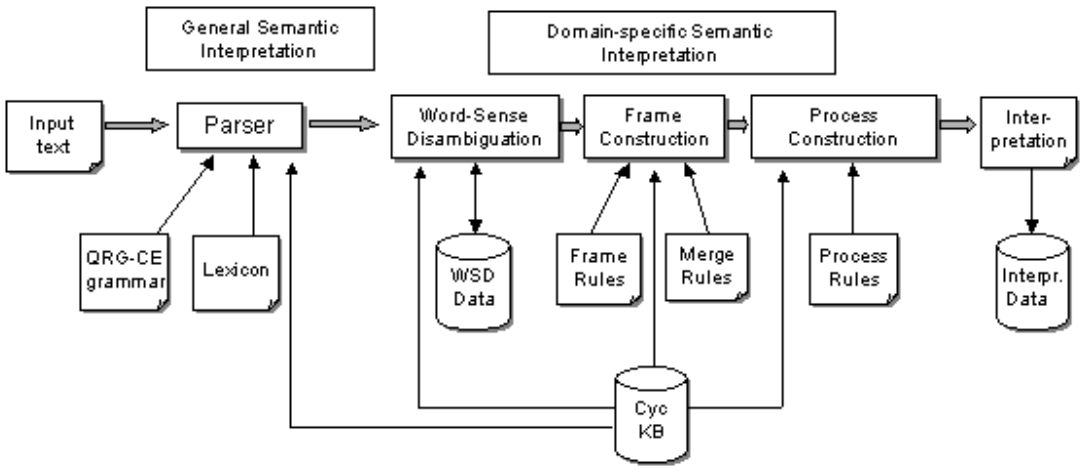

Figure 6.1: Overview of the Semantic Interpretation Process

The second step of the interpretation process resolves these choice sets by employing a word-sense disambiguation algorithm that prefers certain concepts over others, based on supporting domain evidence, selectional restrictions and user preferences. Forward-chaining rules are used to build and merge QP frame structures and create a domain-specific interpretation from the input text. Besides generating semantic interpretations for single sentences, frame information from multiple sentences can be merged to create a paragraph-level interpretation. Figure 6.1 shows an overview of the components involved in the semantic interpretation process.

6.1 The parse-level semantic interpretation

The syntactic analysis and the general semantic interpretation are performed by the modified version of the parser described in chapter 5. Besides the expanded contents of the COMLEX lexicon that are used for the syntactic analysis, the parser makes use of the Cyc knowledge base.[1]

6.1.1 Aligning the parser lexicon with the Cyc knowledge base

The parser takes lexical information from the Cyc Knowledge Base for compiling general, domain-independent semantic information about the input by using the root form of a main lexicon entry to access the Cyc lexicon. Cyc lexicon entries have the form *X-TheWord* where *X* is the capitalized base form of the corresponding root entry from the main lexicon. Other lexical information about an entry in the Cyc lexicon can be found as assertions, as one would expect for any information provided by the knowledge base. Figure 6.2 shows the mapping between the main lexicon entries and the Cyc lexicon for the three entries discussed earlier.

[1] The parser can also operate in a purely syntactic mode, without accessing the Cyc knowledge base to provide semantic information. This mode is particularly useful for developing and debugging grammar rules for new syntactic structures.

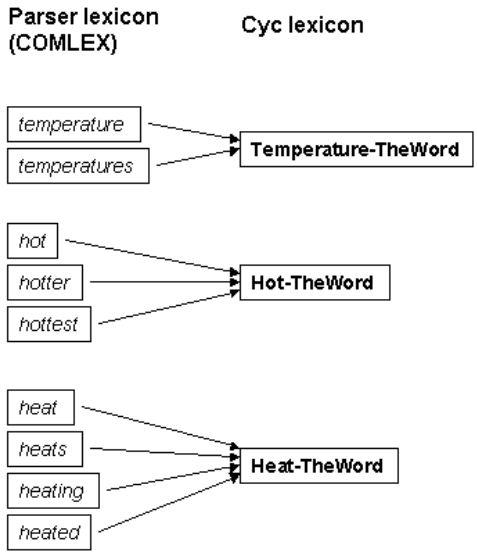

Figure 6.2: Mappings between lexicon entries

The Cyc lexicon entries of the form *X-TheWord*, e.g. Temperature-TheWord, are not the knowledge base concepts corresponding to the parser lexicon entry. It is also important to note that a single Cyc lexicon entry can map to multiple concepts based on their part of speech information in the knowledge base. The semantic information for the lexicon entries is retrieved by applying the correct part of speech to the denotational information stored for each lexicon entry and selecting the appropriate semantic frames. The next section illustrates how the parser uses the lexical information to retrieve semantic information from the Cyc knowledge base. An introduction into the organization of the Cyc lexicon can be found in (Burns & Davis, 1999).

6.1.2 Retrieving semantic information for terminal nodes

The Cyc lexicon entries provide the parser with entry points into the Cyc knowledge base. For each terminal node, the parser retrieves all the concepts and any semantic information attached to these concepts that correspond to the appropriate part of speech. Concepts are linked to the Cyc lexicon entries by the `denotation` predicate. The arguments of this predicate specify the lexicon entry, the part of speech, a word sense number, and an associated Cyc concept. The knowledge base contains the following denotational information for the words *temperature*, *heat*, and *hot*:

```
(denotation Temperature-TheWord CountNoun 0 Temperature)
(denotation Temperature-TheWord CountNoun 1 Fever)
(denotation Temperature-TheWord CountNoun 2 bodyTemperature)

(denotation Heat-TheWord MassNoun 0 ThermalEnergy)
(denotation Heat-TheWord Verb 0 HeatingProcess)
(denotation Heat-TheWord AgentiveNoun 0 HeatingDevice)

(denotation Hot-TheWord MassNoun 0 Hot)
(denotation Hot-TheWord Adjective 1 Hot)
(denotation Hot-TheWord Adjective 2 GoodLooking)
(denotation Hot-TheWord Adjective 3 Hot-Spicy)
```

The parser needs to filter out those denotations that do not correspond to the appropriate lexical category of the terminal node. The possible parts of speech for lexical entries in the Cyc KB are mapped to the lexical categories found in the expanded parser lexicon by the predicate `synonymousExternalConcept`. The following mappings for nouns exist between the lexicon based on the COMLEX 3.1 data and the Cyc lexicon:

```
(synonymousExternalConcept Noun
                           COMLEX31LexicalCategory “noun”)
(synonymousExternalConcept SimpleNoun
                           COMLEX31LexicalCategory “noun”)
(synonymousExternalConcept CountNoun
                           COMLEX31LexicalCategory “noun”)
(synonymousExternalConcept MassNoun
                           COMLEX31LexicalCategory “noun”)
```

After the parser has filtered out all non-matching denotations, it tries to find any semantic information that might be attached to the remaining concepts. This semantic information is specified by POS-specialized `semTrans` predicates. For example, the semantic information for a concept from the noun sense denotation would use the predicate `nounSemTrans`, while `verbSemTrans` would be used for a concept originating from a verb sense denotation. The semantic information itself consists of one or more Cyc expressions that often contain filler keywords. The keywords can be replaced by other expressions, when information from terminal nodes is combined at phrase level. Figure 6.3 summarizes the lookup process for semantic information associated with a terminal node.

Lookup of semantic information

For each word W in part of speech P:

- Determine root form of parser lexicon entry R(W,P)
- Construct (potential) Cyc lexicon entry C from R
- Find denotational information D(C,P) in KB
- Find associated semantic information S(C,P) in KB
- Return denotations D and semantic information S

Figure 6.3: Retrieving semantic information for terminal

Not every concept necessarily has semantic information attached to it, and in some cases we might even find semantic information for a lexical entry without a concept. Although this situation can usually be thought of as a 'hole' in the knowledge base that can be filled by defining an appropriate concept, the parser uses all available semantic information for the terminal nodes. For each node, we need to maintain a list of concepts and a list of semantic information attached to these concepts. If several competing concepts are found for a node, we have to build a choice set. Semantic information that is not attached to any concept will be added as default semantic information for every competing concept in the choice set.

6.1.3 Representations for choices between word senses

Choices for word senses are encoded as a choice set by using the predicate `semChoiceset`. The choice sets contain the concept names as well as additional information about their context, i.e. the discourse variable the choices are bound to, its part of speech, and the originating lexical item. Any semantic information tied to a particular member of the choice set is represented separately as `semTransForChoice` expressions. For example, the following expressions show the choice set representations for the noun *temperature* and the adjective *hot*.[2]

```
(semChoiceset Temperature-TheWord noun temperature202
   (Fever bodyTemperature Temperature))
(semTransForChoice Temperature temperature202
   ((isa temperature202 Temperature)))
(semTransForChoice bodyTemperature temperature202
   ((isa temperature202 bodyTemperature)))
(semTransForChoice Fever temperature202
   ((isa temperature202 Fever)))
```

[2] For the third lexical entry used in the previous examples, the verb *heat*, only one related concept exists in the knowledge base. A choice set representation is therefore not required.

```
(semChoiceset Hot-TheWord adjective hot113
   (Hot-Spicy Hot GoodLooking))
(semTransForChoice GoodLooking hot113
   ((hasAttributes brick113 GoodLooking)
    (hasPhysicalAttractiveness brick113 GoodLooking)))
(semTransForChoice Hot hot113
   ((hasAttributes brick113 Hot)
    (feelsSensation brick113 LevelOfHeatSensation Positive)))
(semTransForChoice Hot-Spicy hot113
   ((hasAttributes brick113 Hot-Spicy)))
```

Ultimately, the semantic information is returned to the parser either in the form of choice sets (if multiple competing concepts exist) or as a simple list of expressions, and stored as the value for sem feature of the terminal node. In addition, the parser creates instances for each concept, using the discourse variable of the node and the concept, e.g. `(isa house3202 House-Modern)`. For certain parts of speech even more information will be added to the node. For example, attributive template expressions will be added for concepts associated with adjectives, e.g. `(hasAttributes :NOUN Hot)`. If the adjective is part of a noun phrase, the keyword `:NOUN` will be substituted by the discourse variable of the noun.

6.1.4 Combining semantic information in phrase nodes

The grammar rules for QRG-CE compose syntactic parse trees in a bottom-up fashion by first combining terminal nodes into phrase nodes, then phrase nodes into other phrase nodes, until a complete sentence structure can be constructed.[3]

When a phrase node for the syntactic parse tree is built, the parser also processes the semantic information from the constituent terminal and phrase nodes. The constituents on the right-hand side of the grammar rule supply various feature information, including their discourse variables and any semantic information stored in their sem features. The parser uses a template to combines this data. Additionally, it will also perform a keyword substitution on the semantic information to replace frame keywords with discourse variable (or different frame keywords, if appropriate).

A common noun phrase (CNP) consisting of an adjective phrase and a noun can be constructed by the following grammar rule:

---

[3] If no complete sentence structure can be build, the parser will return a set of phrase structures plus any leftover terminal nodes.

```
((cnp (agr ?agrn) (var ?varn) (:NOUN ?varn)
      (sem (and ?semap ?semn)))
 -cnp->adjp-noun-
 (adjp (sem ?semap))
 (head (noun (agr ?agrn) (var ?varn) (sem ?semn))))
```

On the right-hand side, the semantic information from the adjective phrase and the noun are bound to the variables `?semap` and `?semn`, the discourse variable of the noun to `?varn`, and the agreement information to the variable `?agrn`. These variables are then used on the left-hand side in the construction of the common noun phrase. The common noun phrase will take the agreement information and the discourse variable from the noun constituent. The semantic information from the adjective phrase and the noun will be combined in a conjunction. And finally, every occurrence of the keyword `:NOUN` in the semantic information will be replaced by the discourse variable of the noun. For example, if the semantic information of adjective phrase contains a template expression `(hasAttributes :NOUN Hot)` and the value of `?varn` is `brick32`, the semantic information of the new phrase node will contain the expression `(hasAttributes brick32 Hot)`.

Similarly, a verb phrase that takes a simple verb phrase and a noun phrase as its right-hand side constituents can be constructed by the following rule:

```
((vp (var ?varv) (agr ?agr) (:OBJECT ?varnp)
     (sem (and ?semv ?semnp)))
 -vp->svp-np-
 (head (svp (subcat np) (agr ?agr) (var ?varv) (sem ?semv)))
 (np (var ?varnp) (sem ?semnp)))
```

The noun phrase can be treated an object and the substitution on the left-hand side will replace any occurrence of the `:OBJECT` keyword in the combined semantic information with the discourse variable of the noun phrase. The substitution is not a condition for the construction of the phrase. If no suitable keyword is found in the semantic information, the substitution request is simply ignored.

6.1.5 Processing semantic information from parse trees

When the parser has completed the analysis and returns the parse trees for the input sentence, the **sem** feature of the top-most nodes contains the combined semantic information from all constituent nodes. For a successful parse, these root nodes consist of the set of sentence-level nodes, each containing a particular syntactic interpretation of the sentence. For an unsuccessful partial parse, the root nodes consist of a set of phrase-level and leftover terminal nodes.

The semantic interpreter handles these two possible outcomes of the parsing step differently. For complete sentence-level parse trees, the semantic interpreter extracts the information stored in the **sem** feature of each tree and processes each interpretation individually. For partial parses, the semantic interpreter extracts the semantic information from all phrase and terminal nodes in the top-level set and combines this information.

In the next step, the semantic interpreter removes all expressions that contain incomplete or erroneous information. For example, expressions that still contain unreplaced frame keywords or unusable knowledge base variables will not be useful for the subsequent interpretation steps and are therefore excluded. Unreplaced frame keywords are pieces of template information that could not be filled in by the parser during the generation of the general semantic interpretation. Since these templates expressions are specified as semantic information, the template might not fit a particular syntactic interpretation. Similarly, some information stored in the knowledge base contains variable information used in other reasoning tasks, but which is irrelevant for a particular sentence. In both cases, the general interpretation would contain expressions with unresolved frame keywords that need to be removed. Once this 'clean-up' step is complete, the semantic interpreter attempts to find the most relevant choices for lexical items that could be mapped to different concepts in the knowledge base.

6.2 Evidence-based Word-Sense Disambiguation

Natural language understanding systems that use a lexicon or a background knowledge base of conceptual information are mostly likely confronted with the fact that many words have more than one distinct meaning and can appear in various parts of speech. Lexical and semantic ambiguity can only be avoided if the lexicon (or the knowledge base) is tightly controlled and allows only one particular semantic interpretation for each word. As described in chapter 5, classic controlled languages like Basic English (C. K. Ogden, 1937) impose a strict one-to-one correspondence between parts of speech and word meanings on all entries in the lexicon to avoid any kind of word-sense ambiguity. The great advantage of this approach is that semantic information associated with each entry in the lexicon is determined by a simple lookup operation. Allowing only one meaning and one part of speech for each word in the lexicon makes the language very restrictive and severely limits its expressiveness. For this reason, word-sense disambiguation becomes a necessary 'intermediate' task (Wilks & Stevenson, 1996) in most natural language processing systems, i.e. word-sense disambiguation is necessary for accomplishing other, more important tasks such as the semantic interpretation of a sentence, rather than being a goal of the NL system by itself.

The lexicon and the knowledge base used in our system are not tightly controlled and allow multiple entries for a single word in a particular part of speech. When the knowledge base is queried for semantic information about a terminal node, the results are often ambiguous such that two or more competing concepts are retrieved for that node.

When the query to the knowledge base produces multiple concepts, the system needs to disambiguate between the different word senses. The parser performs a syntactic disambiguation step when it constructs the parse tree based on its grammar rules. For example, if the grammar accepts the sentence *'The house is on the hill.'* the word *house* is treated as the head of a noun phrase. In this case, the system can discard all information for the verb sense of the word *house* and only keep the semantic information associated with the noun sense.

Even though some conflicting information can be ruled out by using syntactic constraints, there are often ambiguous entries for the same part of speech of a word. In the present example, there are four different senses for the noun *house* left that need to be disambiguated. A set of semantic constraints is necessary to allow us to distinguish between these different word senses.

A straightforward approach to semantic disambiguation is to frame the choice between different word senses as a constraint satisfaction problem (CSP) and eliminate all non-fitting senses based on constraints provided by contextual information (Mellish, 1985). (Allen, 1995) describes a simple algorithm for such an approach. Employing a constraint satisfaction algorithm for word-sense disambiguation will only work sufficiently well if the contextual information is free of inconsistencies. Any erroneous data will lead to imprecise or even wrong constraints, and these will eventually lead to the elimination of desired word senses. CSP-based approaches might work well for small, handcrafted lexicons and knowledge bases. In our case, using third-party provided resources such as the COMLEX lexicon and the contents of the Cyc knowledge base, we have to assume inconsistencies such as missing entries, non-aligned argument structures and erroneous part of speech information. For this reason we use an evidence-based approach to word-sense disambiguation that collects and weighs various types of evidence supporting a word sense.

6.2.1 Evidence-based word-sense disambiguation

The system uses an evidence-based approach, such that for each set of choice between word senses a number of evidential tests are applied to each candidate within the set. Weights are assigned for each test, with the actual values of the weights depending on general and task-specific criteria and their evidential relevance. The word sense with the highest score is then picked as the best available choice. In the following section,

we will discuss the evidential tests that are used for resolving ambiguous choice sets of word senses.

The tests fall into four major categories: tests for task-specific evidence, tests for contextual restrictions, tests based on preferences in the knowledge base, and tests for user preferences. Task-specific evidence and user preferences should be regarded as the strongest and most valuable evidence, while preferences for particular KB content should be counted only as a very weak contribution.

6.2.1.1 Task-specific evidence

The task-specific tests are primarily based on the relevance of a concept for the domain in which the system is operating. In our particular case, the task-specific tests are based on qualitative physics and look for the following types of evidence:

- quantity types, such as *Temperature* or *Mass*?
- concepts related to a quantity type, such as *Warm* or *Cold*?
- units, such as *Liter* or *Kilogram*?
- physical processes, such as *HeatingProcess*?
- concepts related to a physical process, such as *IncreaseEvent*?
- concepts marked as a domain-specific term?

If a concept meets one or more of these criteria, it is assigned a weighted score that indicates the relevance of the evidence. For example, information about a quantity type is weighted higher than information that a concept is just related to a quantity type. Similarly, a concept that denotes a physical process is more relevant than just some process-related piece of information, i.e. a concept that appears in the semantic information attached to a concept denoting a physical process. The concept associated with the `objectMoving` in a `TranslationFlow` is process-related, while the `Translation-Flow` denotes a physical process.

6.2.1.2 Contextual constraints

The second category of evidence relevant for the disambiguation of concepts is based on contextual constraints such as selectional restrictions and the interactions between words in a sentence. For example, the semantic information associated with a word might require that the arguments meet certain requirements.

The semantic information associated with words in a sentence can produce predicates that impose certain restrictions on their arguments. The idea of using selectional restrictions goes back to (Katz & Fodor, 1963) and has since then been used in many

natural language processing systems (Hirst, 1987; Lehnert, Dyer, Johnson, Yang, & Harley, 1983; Wilks, 1975a).

The semantic interpreter in our system retrieves all expressions in which the choice set variable is used, retrieves the selectional restrictions for the particular slot in those expressions, and then tests each member of the choice set. The evidence for the selectional restriction test is ranked by the highest to the lowest relevance:

- the concept *directly matches* the slot restrictor
- the concept is an *instance* of the slot restrictor
- the concept is a *subset* of the slot restrictor

For example, the binary predicate `emptiesInto` uses the following restrictions on its arguments:

```
(emptiesInto ?river ?water)
(arg1Isa emptiesInto Stream)
(arg2Isa emptiesInto BodyOfWater)
```

The first argument should be a `Stream` (or a subcollection or instance of it) while the second argument is a `BodyOfWater`. The following three sentences will match the slot restrictions of the predicate `emptiesInto`:

(1) The *stream* flows into the ocean.
(2) The *Mississippi* flows into the ocean.
(3) The *river* flows into the ocean.

In (1) the noun *stream* will resolve into the concept `Stream`, which directly matches the slot restrictor for the first argument. In (2) the proper noun *Mississippi* corresponds to the concept `MississippiRiver`, which is an instance of `Stream` and fulfills the slot restrictions, although it is not a 'good' direct match as the choice in (1). In sentence (3), the noun *river* is associated with the concept `River`, which is a subcollection of `Stream`. It is an acceptable fit for the first argument, but as a subcollection it is not as close to the restrictor as the choices in (1) and (2).[4]

The semantic interpreter just assigns weights instead of eliminating information that does not fit slot restrictions. Parts of the background information in our knowledge base contain potentially incomplete and inconsistent information. The choice set

---

[4] The test for subsets is a potentially weak contribution of evidence. Often slot restrictors are very general, such as `SomethingExisting` or even `Thing`, the topmost collection in the 30,000+ set of collections in the KB.

resolution algorithm acknowledges this fact by using weights and making the selectional restrictions just one part of the evidence for a particular choice. For example, the noun *garbag*e in sentence (4) might resolve into the concept `Trash`, but it is neither a `Stream` nor an instance or subcollection of it.

(4) The *garbage* flows into the ocean.

The use of strong constraints would force us to eliminate this choice from a choice set, because it does not fit the slot restrictions of an expression in which it is used, potentially resulting in no fitting concept at all.

6.2.1.3 Preferences based on KB contents

The third class of evidence for the preference of concepts is based on the contents of the knowledge base. Our subset of the Cyc knowledge base contains information about specializations, lexical information about word senses, and the preference for concepts in natural language generation that can be used as weak evidence. The preferences encoded in the knowledge base are subjective data and usually reflect the ideas and views of the knowledge base developers. Therefore, this kind of information should not be regarded as high-quality strong evidence but only as a minor contribution in a comprehensive evaluation.

- **Preference for specialization**
  Information about specialization is also used for the resolution of choice sets. In general, our algorithm prefers specific concepts over their more general competitors. For example, if a choice set contains the concepts `Hill` and `Hill-Topographical-Generic`, `Hill` is preferred over its immediate parent collection `Hill-Topographical-Generic`. If a concept is a specialization of one of its competitors, the more specific concept will pass the preference test for specialization.

- **Concepts preferred in NL generation**
  The knowledge base contains information that specifies which lexical entries are the preferred in generating natural language text from knowledge base contents. These preferences usually exist for the most commonly used lexical entry connected to the concept. This fact is exploited as weak evidence that these concepts might be rated slightly more relevant than their alternatives by the developers of the knowledge base.

- **Preference for certain parts of speech**
  A similar approach is used to give preference to concepts associated with a particular part of speech. For example, the lookup for noun *heat* results in the two concepts `ThermalEnergy` and `HeatingDevice`. `ThermalEnergy` is the concept connected to the mass noun sense of heat, while the `HeatingDevice` is used for agentive noun sense. The concept for the mass noun sense is a better fit for the word *heat* and will get preference in form of a higher score over the agentive noun. Similarly, for the noun *truck* the knowledge base contains the concepts `Truck` and `TruckDriver`. In this case, the count noun should also get a higher score than the agentive noun.

6.2.1.4 Learned user preferences

The fourth type of evidence is based on user preferences for particular concepts. Our system allows choice sets to be resolved in two different ways: (a) by using an automatic mode in which a number of heuristics are used to collect evidence for the best possible choice, and (b) by using an interactive mode, in which a user is manually selecting the best fitting concept.

In interactive mode, the semantic interpreter generates a list of ambiguous concepts and asks the user to select the most appropriate concept. The choice of the user is then recorded in a database for word-sense disambiguation information. The data includes the lexical word, the selected concept, and a timestamp. The timestamp is used to distinguish between multiple choices for the same word/concept pair. Multiple entries for the same word/concept pair will make choice more relevant, i.e. a concept has that been selected 10 times as the best fit for a lexical word will be treated as more relevant as a concept that has been selected only once. The word-sense disambiguation database is separated from the background knowledge base, so that the statistical information from the manual word-sense disambiguation process will not bloat the background knowledge base. Moreover, different word-sense databases can be used for different tasks, domains, and users. The manual word-sense disambiguation should be regarded as a training tool and for aiding the resolution of choice sets that cannot be successfully processed in the automatic, heuristics-based mode.

This data should be regarded as strong evidence since it is based on interactively collected data. Training the system by manually selecting the best fitting choice for a set of ambiguous concepts is a tedious process, so the results should be treated as very valuable evidence for the automatic choice set resolution.

6.2.2 Evaluation of evidence

The results of the evidential tests for each concept are accumulated as a normalized score. The scores for each set of competing word senses are sorted and the highest scoring concept will be picked as the most appropriate choice.

The individual score S for each concept C_i are computed from the results of n evidential tests $T_1 \ldots T_n$ and their associated weights $W_{T1} \ldots W_{Tn}$. If the concept C_i fails to pass a test T_j, the associated weight $W_{Tj}(C_i)$ is 0.

$$\boldsymbol{S(C_i)} = \sum_{j=1}^{n} W_{Tj}(Ci) \qquad \textbf{(1)}$$

As soon as all individual scores for the competing concepts in a set have been computed, the system normalizes the scores. The normalized score NS for a concept C_i is

$$\boldsymbol{NS(C_i)} = \frac{S(Ci)}{\sum_{j=1}^{m} S(Cj)} \qquad \textbf{(2)}$$

The individual scores reflect the relevance of the evidence provided by concept. The highest scoring concept is subsequently selected as the best choice.

Depending on task and application specific requirements the scores can be weighted and adjusted. For example, the task-specific evidential tests can be completely disabled to remove a bias for a particular interpretation of concepts. The preference for any domain concepts in the resolution of choice sets can be turned off, resulting in an unbiased disambiguation process that only depends on general restrictions and preferences in the knowledge base.

6.2.3 Representations of resolved choice set information

The word-sense disambiguation process uses the predicates `supportForChoice` for individual pieces of evidence and the predicate `scoreForChoice` for the normalized score of each choice. For example, the choice set resolution for the noun ‘heat’ can consists of the following expressions.

```
(supportForChoice ThermalEnergy heat151699
        (genls ThermalEnergy PhysicalQuantity) quantitytype)
(supportForChoice ThermalEnergy heat151699
        (objectMoving flow151714 heat151699) slotsubset)
```

```
(supportForChoice ThermalEnergy heat151699
        (posForms Heat-TheWord MassNoun) pos-preference)
(supportForChoice ThermalEnergy heat151699
        (numberOfUserSelections ThermalEnergy 4)
                                        user-preference)

(supportForChoice HeatingDevice heat151699
       (primaryObjectMoving flow151714 heat151699) slotsubset)
(supportForChoice HeatingDevice heat151699
       (objectMoving flow151714 heat151699) slotsubset)

(scoreForChoice HeatingDevice heat151699 0.15)
(scoreForChoice ThermalEnergy heat151699 0.85)
```

After the choice set has been resolved, the best choice will be marked by the `bestChoice` predicate.

```
(bestChoice Heat-TheWord noun ThermalEnergy heat151699
       (TheList HeatingDevice ThermalEnergy))
```

Since some of the steps in the disambiguation process rely on specific information from the background knowledge base, we do not claim our method is a general theory of word sense disambiguation. It is a pragmatic approach to disambiguate competing concepts associated with lexical items by using task-specific constraints imposed by the domain, general information from the background knowledge base, and user preferences from interactive training sessions.

The use of an evidence-based approach instead of hard constraints allows us to deal with inconsistencies in the underlying knowledge base. For future extension of this work, this approach will enable us to use the contents of the knowledge base for metaphorical interpretations of concepts. Selectional restrictions used in expressions are usually intended to work for literal interpretations only. They do not work particularly well for metaphorical interpretations, unless the restrictions are so loose that almost any concept fit them. Hard constraints would eliminate metaphorical senses for a word because they violate the selectional restrictions. An evidence-based approach will consider metaphorical uses of a word if enough other information supports that interpretation. For example, for the sentence ‘His anger boils over’ the selectional restrictions found in the semantic information for the verb ‘boil’ require the subject of the sentence to be interpreted as a substance. By default, the concept associated with the noun ‘anger’ would not meet this requirement and it would be rejected as a potential choice if hard constraints were used. However, the knowledge base could contain additional information such as user and task-specific preferences for the interpretation of anger as a substance. In this case, the supporting evidence would be used to allow the metaphorical interpretation as a valid choice. We will come back to these issues and address future work in chapter 8.

6.3 Building QP frames

After the word-sense disambiguation process has resolved any potentially ambiguous concepts in the current interpretation data, the next step in the semantic interpretation process is to capture QP-related information in a set of frame structures. This section discusses the forward-chaining LTRE rules (Forbus & de Kleer, 1993) that guide the construction of QP frames.

6.3.1 Quantity Frames

Quantity frames are constructed for expressions that describe relationships between objects and for attributes associated a particular object. The defining factor is whether one of the objects or the attribute is a quantity type or at least related to a quantity type. For example, the following rule looks for an attribute associated with some individual and tests via its discourse variable whether the attribute is a quantity type. If these conditions are met, the rule will construct a new quantity frame that contains the individual, its quantity type, and the attribute as a value. The expression `(hasAttributes brick32 Hot)` will lead to the construction of a Quantity frame for the temperature of the brick, assuming that `Hot` is a subcollection of `Temperature`.

```
(rule ((:true (hasAttributes ?entity ?attr)
              :var ?ha)
       (:true (isa ?attr ?qtype)
              :var ?isqtype
              :test (quantity-type-p ?qtype)))
      (let ((?qframe (make-frameid 'QuantityFrame)))
        (rassert!
         (:implies (:and ?ha ?isqtype)
                   (:and (isa ?qframe QuantityFrame)
                         (entity ?qframe ?entity)
                         (quantityType ?qframe ?qtype)
                         (quantityValue ?qframe ?attr)
```

Quantity frames can also be based on possessive relationships, expressed by phrases like 'the temperature of the brick'. The following rule checks whether the thing possessed by an individual is associated with a quantity type and instantiates the corresponding quantity frame.

```
(rule ((:true (possessiveRelation ?owner ?thing)
              :var ?pr)
       (:true (isa ?thing ?qtype)
              :var ?isqtype
              :test (quantity-type-p ?qtype)))
      (let ((?qframe (make-frameid 'QuantityFrame)))
```

```
        (rassert!
         (:implies (:and ?pr ?isqtype)
                   (:and (isa ?qframe QuantityFrame)
                         (entity ?qframe ?owner)
                         (quantityType ?qframe ?qtype))))))
```

There are cases in which the thing ‘owned’ by the individual is not a direct reference to a quantity type. For example, in the noun phrase ‘*the water of the source*’ the concept for *water* is not `PhysicalQuantity` and the previous rule would fail to recognize that the amount of water at the source does indeed constitute a physical quantity. The following rule allows the construction of a Quantity frame for possessive relations as long as the thing possessed is a mass noun. It will use an alternate representation for the quantityType slot that expresses the concepts associated with these mass nouns as amounts.

```
(rule ((:true (possessiveRelation ?owner ?thing)
              :var ?pr)
       (:true (isa ?thing ?col)
              :var ?iscol
              :test (and (not (quantity-type-p ?col))
                         (not (event-p ?col))
                         (mass-noun-p ?col))))
      (let ((?qframe (make-frameid ‘QuantityFrame)))
        (rassert!
         (:implies (:and ?pr ?iscol)
                   (:and (isa ?qframe QuantityFrame)
                         (entity ?qframe ?owner)
                         (quantityType ?qframe (AmountFn ?col)))))))
```

Physical quantities can also be location-based or expressed as containments. Again, the quantity type can be directly referenced or treated as an amount of some mass noun. The phrases ‘*the pressure in the cylinder*’ or ‘*the water at the bottom*’ are examples for these kinds of descriptions. The following rule captures information from containment relations.

```
(rule ((:true (in-UnderspecifiedContainer ?stuff ?container)
              :var ?in)
       (:true (isa ?stuff ?qtype)
              :var ?isqtype
              :test (and (quantity-type-p ?qtype)
                         (mass-noun-p ?qtype))))
      (let ((?qframe (make-frameid ‘QuantityFrame)))
        (rassert!
         (:implies (:and ?in ?isqtype)
                   (:and (isa ?qframe QuantityFrame)
                         (entity ?qframe ?container)
                         (quantityType ?qframe ?qtype))))))
```

6.3.2 Changes in Quantities

Forward-chaining LTRE rules are also used to identify the sign of derivative in Quantity Frames. Verbs that denote changes in quantities such as *increase, rises*, or *drops* are tied to the collections `IncreaseEvent` and `DecreaseEvent`. The semTrans information for these events uses the predicate `objectActedOn` to refer to the quantity type, e.g. the temperature in the sentence *'The temperature of the water increases.*' The following rule sets the signOfDerivative slot of an existing Quantity frame to `Positive` if the quantity type is involved in an `IncreaseEvent`.

```
(rule ((:true (isa ?event IncreaseEvent)
              :var ?ie)
       (:true (objectActedOn ?event ?thing)
              :var ?oao)
       (:true (isa ?qframe QuantityFrame)
              :var ?qf)
       (:true (quantityType ?qframe ?qtype)
              :var ?qt)
       (:true (isa ?thing ?qtype)
              :var ?isqt))
      (rassert!
       (:implies (:and ?ie ?qf ?oao ?qt ?isqt)
                 (signOfDerivative ?qframe Positive))))
```

6.3.3 Quantity Transfer frames

Quantity Transfer frames are important for capturing information about changes in quantities based on transfer-related events expressed by verbs like *flow* or *move*. For example, the verb *flow* uses the prepositions *from* and *to* in the Transfer frame to indicate the source and destination of the flow, while in the Motion frame the same prepositions would be treated as locations. Although the sentences *'Heat flows from the brick to the room.*' and *'The particle flows from the tank to the nozzle.*' are from two different domains, their underlying frame structure is similar.

Events are tied to QP frames by using the predicate `relatedToQPFrame`. A `Translation-Flow` as one of the concepts associated with the verb *flow* and a `MovementEvent` associated with the verb *move* are both related to the Quantity Transfer Frame by this predicate. If the semantic interpreter finds events associated with Quantity Transfer frames, it will add the appropriate definition to the current interpretation.

Once a frame associated with an event is known, further information such as the roles for the frame can be retrieved from the knowledge base to fill the slots of the Quantity

Transfer frame. The role information uses the `supportsRoleInFrame` predicate and allows a mapping from generic Cyc predicates for role relations to the roles in a particular QP frame. The `supportsRoleInFrame` expressions create an additional layer of abstraction to allow many-to-one mappings and the use of a minimum number LTRE rules.

Note that the specialized role relations are not the predicates for the frame slots but provide information for constructing the frame slot information. A good example is the sourceOfTransfer slot for the QuantityTransfer frame. For an expression found in the general semantic interpretation data, such as `(from-generic flow123 brick456)`, we cannot use `brick456` as an argument for the predicate `sourceOfTransfer`, since it expects a quantity frame in the second argument position, not an entity. To construct a quantity frame from the `sourceOfTransfer` expression, we need to combine the information in the `from-generic` expression with information about a quantity type participating in the transfer event. A rule for constructing the Quantity frame for the source in a Quantity Transfer frame is shown below. Similar rules are used for the destination of transfer events.

```
(rule ((:true (isa ?qtframe QuantityTransferFrame)
              :var ?qt)
       (:true (transferredStuff ?qtframe ?stuff)
              :var ?transfer)
       (:true (isa ?stuff ?qtype)
              :var ?isqtype
              :test (and (quantity-type-p ?qtype)
                         (mass-noun-p ?qtype)))
       (:true (sourceLocOfTransfer ?qtframe ?srcloc)
              :var ?src))
      (let ((?qframe (make-frameid 'QuantityFrame)))
        (rassert!
         (:implies (:and ?qt ?transfer ?isqtype ?src)
                   (:and (isa ?qframe QuantityFrame)
                         (entity ?qframe ?srcloc)
                         (quantityType ?qframe ?qtype)
                         (sourceOfTransfer ?qtframe ?qframe))))))
```

The transfer rate between two quantities is (in most cases) not explicitly mentioned, unless a numeric value about the flow rate is known and specified, as in '*The water flows from the tank to the cylinder at a rate of 9.67 liters per minute.*' If no rate is mentioned in conjunction with an event, a default Quantity frame for the rate is constructed by a separate rule. The entity of the Quantity frame for a rate is the discourse variable of the transfer event itself. The rate is treated as an internal quantity directly associated with the transfer.

6.3.4 Direct Influence frames

The information captured by QuantityTransfer frames can be used to instantiate at least one DirectInfluence frame.[5] The source and the destination of a QuantityTransfer frame are the constrained quantity of a DirectInfluence frame, while the quantity for the transfer rate is the constrainer of the DirectInfluence frame. The sign of the Influence frame will be positive for the destination of the transfer (I+) and negative for the source (I-). The following rule is used for building a DirectInfluence frame for the source of the transfer.

```
(rule ((:true (sourceOfTransfer ?qtframe ?src)
              :var ?transfersrc)
       (:true (rateOfTransfer ?qtframe ?rate)
              :var ?transferrate))
      (let ((?diframe (make-frameid 'DirectInfluenceFrame)))
        (rassert!
         (:implies (:and ?transfersrc ?transferrate)
                   (:and (isa ?diframe DirectInfluenceFrame)
                         (constrained ?diframe ?src)
                         (constrainer ?diframe ?rate)
                         (sign ?diframe Negative))))))
```

6.3.5 Indirect Influence frames

The parser gathers information about qualitative proportionalities while constructing a general semantic interpretation, assisted by QRG-CE grammar rules for capturing the semantic pattern that are commonly used for expressing qualitative proportionalities. The domain-specific semantic interpreter processes all these expressions to construct IndirectInfluence frames.

The difficulty is to find the links between the events referenced in qualitative proportionalities and the quantities involved in the events, since Quantity frames can be constructed independently from a particular events. The interpretation of the noun phrase 'a large car' results in the instantiation of a Quantity frame for the entity *car*, the quantity type *size* and the value *large*. The Quantity frame does not have to be explicitly tied to an event.

The following rule uses the predicate `objectedActedOn` in `IncreaseEvent` and `DecreaseEvent` information to identify the quantities involved in an IndirectInfluence frame.

---

[5] If the QuantityTransfer frame is complete and includes the quantities for the source and the destination of the transfer, two DirectInfluence frames will be constructed.

```
(rule ((:true (qpropEvent ?ev1 ?ev2)
              :var ?qprop)
       (:true (isa ?ev1 ?event1)
              :var ?isev1)
       (:true (isa ?ev2 ?event2)
              :var ?isev2)
       (:true (objectActedOn ?ev1 ?thing1)
              :var ?oao1)
       (:true (objectActedOn ?ev2 ?thing2)
              :var ?oao2)
       (:true (isa ?thing1 ?qtype1)
              :var ?isqt1)
       (:true (isa ?thing2 ?qtype2)
              :var ?isqt2)
       (:true (quantityType ?qf1 ?qtype1)
              :var ?qt1)
       (:true (quantityType ?qf2 ?qtype2)
              :var ?qt2
              :test (not (eql ?qf1 ?qf2)))
       (:true (signOfDerivative ?qf1 ?sign1)
              :var ?s1)
       (:true (signOfDerivative ?qf2 ?sign2)
              :var ?s2))
      (let ((?iiframe (make-frameid 'IndirectInfluenceFrame))
            (?sign (sign-for-qprop ?sign1 ?sign2)))
        (rassert!
         (:implies (:and ?qprop ?oao1 ?oao2 ?qt1 ?qt2 ?isqt1 ?isqt2
                         ?s1 ?s2 ?isev1 ?isev2)
                   (:and (isa ?iiframe IndirectInfluenceFrame)
                         (constrained ?iiframe ?qf1)
                         (constrainer ?iiframe ?qf2)
                         (sign ?iiframe ?sign))))))
```

For example, the sentence *'As the pressure in the cylinder rises, the flow rate of the water increases.'* uses a syntactic pattern for indirect influences (Chapter 3). It contains information about two quantities (the pressure in the cylinder, and the flow rate of the water). The changes in the quantities would be interpreted as two IncreaseEvents, in which the cylinder and the water are involved. The rule combines information from the two Quantity frames as well as the IncreaseEvents to construct an IndirectInfluence frame.

6.3.6 Ordinal Relation frames

Ordinal relation frames are constructed whenever two quantities are compared. As discussed in chapter 3, such comparisons can be expressed directly or indirectly. A direct comparison explicitly mentions the two quantities and the relation between them. Sentences like *'The length of the car is greater than the length of the truck.'* or

'The tree is taller than the house.' are direct comparisons. The former uses a quantity-neutral adjective while the comparative in the latter is quantity-specific. Both sentences name the compared entities, i.e. the compared item (the 'comparer') and the reference (the 'comparee').

The following rule builds an OrdinalRelation frame from the comparison information in sentence with quantity-neutral constructions. The rule triggers search for possessive relations between the entities and their quantity types. This information is also used in the construction of the actual Quantity frames, as described earlier. Furthermore, the relationship between the two quantities is determined by an external function that queries the knowledge base for polarity and ordering information.

```
(rule ((:true (isa ?qf1 QuantityFrame)
              :var ?q1)
       (:true (isa ?qf2 QuantityFrame)
              :var ?q2
              :test (not (eql ?qf2 ?qf1)))
       (:true (entity ?qf1 ?entity1)
              :var ?ent1)
       (:true (entity ?qf2 ?entity2)
              :var ?ent2)
       (:true (quantityType ?qf1 ?qtype)
              :var ?qt1)
       (:true (quantityType ?qf2 ?qtype)
              :var ?qt2)
       (:true (isa ?qtvar1 ?qtype)
              :var ?qtv1)
       (:true (isa ?qtvar2 ?qtype)
              :var ?qtv2)
       (:true (isa ?compevent ComparisonEvent)
              :var ?cevent)
       (:true (comparer ?compevent ?qtvar1)
              :var ?cr)
       (:true (comparee ?compevent ?qtvar2)
              :var ?ce)
       (:true (comparativeRelation ?compevent ?comprel)
              :var ?crel)
       (:true (degreeInformation ?comprel ?degree ?thing)
              :var ?dgr)
       (:true (possessiveRelation ?entity1 ?qtvar1)
              :var ?pr1)
       (:true (possessiveRelation ?entity2 ?qtvar2)
              :var ?pr2))
      (let ((?orframe (make-frameid 'OrdinalRelationFrame))
            (?relation (comparative-relation-for ?degree)))
        (rassert!
         (:implies (:and ?q1 ?q2 ?cr ?ce ?pr1 ?pr2
                         ?qt1 ?qt2 ?qtv1 ?qtv2 ?ent1 ?ent2 ?cevent)
                   (:and (isa ?orframe OrdinalRelationFrame)
```

```
                        (quantity1 ?orframe ?qf1)
                        (quantity2 ?orframe ?qf2)
                        (relationBetweenQuantities ?orframe
                                                   ?relation))))))
```

The rule for quantity-specific direct comparisons cannot use information from already existing quantity frames as trigger. Instead, it has to create the quantity frames from the information comparison itself. The quantity type is determined by the comparative, the entities are the two participants of the comparative event.

```
(rule ((:true (isa ?compevent ComparisonEvent)
              :var ?cevent)
       (:true (comparer ?compevent ?comparer)
              :var ?cr)
       (:true (comparee ?compevent ?comparee)
              :var ?ce)
       (:true (comparativeRelation ?compevent ?comprel)
              :var ?crel)
       (:true (degreeInformation ?comprel ?degree ?thing)
              :var ?dgr)
       (:true (isa ?thing ?qtype)
              :var ?isqt
              :test (or (quantity-type-p ?qtype)
                        (related-to-quantity-type-p ?qtype))))
      (let ((?orframe (make-frameid 'OrdinalRelationFrame))
            (?qframe1 (make-frameid 'QuantityFrame))
            (?qframe2 (make-frameid 'QuantityFrame))
            (?relation (comparative-relation-for ?degree)))
        (rassert!
         (:implies (:and ?cevent ?cr ?ce ?crel ?dgr ?isqt)
                   (:and (isa ?qframe1 QuantityFrame)
                         (entity ?qframe1 ?comparer)
                         (quantityType ?qframe1 ?qtype)
                         (isa ?qframe2 QuantityFrame)
                         (entity ?qframe2 ?comparee)
                         (quantityType ?qframe2 ?qtype)
                         (isa ?orframe OrdinalRelationFrame)
                         (quantity1 ?orframe ?qframe1)
                         (quantity2 ?orframe ?qframe2)
                         (relationBetweenQuantities ?orframe
                                                    ?relation))))))
```

Ordinal Relation frames can also be constructed from implicit comparisons based on the value information in Quantity frames. If the value information is numeric, figuring the ordinal relation between two frames of the same quantity type is easy. The more interesting case can be found for Quantity frames that contain only symbolic values. For example, a sentence can mention a *hot brick* and the *cool ground*.

The rule shown below looks for Quantity frames with symbolic value information. The order between the two frames is determined by querying the knowledge base for partial ordering information the symbolic values.

```
(rule ((:true (isa ?qf1 QuantityFrame)
              :var ?q1)
       (:true (isa ?qf2 QuantityFrame)
              :var ?q2
              :test (not (eql ?qf2 ?qf1)))
       (:true (quantityType ?qf1 ?qtype1)
              :var ?qt1)
       (:true (quantityType ?qf2 ?qtype2)
              :var ?qt2
              :test (same-dimension-p ?qtype1 ?qtype2))
       (:true (quantityValue ?qf1 ?qvalue1)
              :var ?qv1
              :test (symbolic-value-p ?qvalue1))
       (:true (quantityValue ?qf2 ?qvalue2)
              :var ?qv2
              :test (symbolic-value-p ?qvalue2)))
     (let ((?orframe (make-frameid 'OrdinalRelationFrame))
           (?relation (symbolic-relation-between ?qvalue1 ?qvalue2)))
       (rassert!
        (:implies (:and ?q1 ?q2 ?qt1 ?qt2 ?qv1 ?qv2)
                  (:and (isa ?orframe OrdinalRelationFrame)
                        (quantity1 ?orframe ?qf1)
                        (quantity2 ?orframe ?qf2)
                        (relationBetweenQuantities ?orframe
                                                   ?relation))))))
```

6.4 Merging QP frames

An essential part of semantic interpretations for natural language text is the use of discourse variables to distinguish between semantic entities such as individuals, events, and values. The identification of these entities is essential for tasks such as creating a consistent semantic interpretation of a sentence or paragraph. It is expected that references to the same entity use the same discourse variable. Similarly, two different discourse variables denote two distinct semantic entities, even if their lexical form is the same.

A simple method to maintain information about discourse variables would be the use of a lookup table for already introduced entities. If the same word or a compatible pronoun appears within a defined window, the lookup algorithm could assign it the previously used variable. In sentence (5) the same discourse variable would be used for both occurrences of the word *brick*. The strategy would even work for entities in

subsequent sentences such as (6) and (7), as long as the occurrences of the same word appear within the lookup window.

(5) As the brick is heated, the temperature of the brick increases.
(6) The brick is heated.
(7) The temperature of the brick increases.

Unfortunately, this strategy is too simple and can introduce more semantic ambiguity than it actually resolves. For example, in sentence (8) the word *temperature* appears twice, although in two different prepositional phrases.

(8) The temperature of the brick is higher than the temperature of the ground.

The two temperatures are different discourse entities, i.e. *the temperature of the brick* and *the temperature of the ground*. The use of a simple lookup strategy would nevertheless assign the same discourse variable to both temperatures. The result is that a subsequent domain-specific semantic interpretation process would assign the same temperature to the brick and the ground.

(9) The heat flows from the brick to the ground.

Similarly, an entity can be mentioned only once in a sentence but could be referred to by two different discourse variables in a semantic interpretation. The heat in (9) can be interpreted as the quantity type associated with the brick and the ground. The heat of the ground should be distinguished from the heat of the brick, even though the quantity type is referred to by the same discourse variable. This is not a problem for the interpretation process, because the semantic interpreter uses different Quantity frames to distinguish the two heat quantities.

The parser does not use a lookup table or any other, more sophisticated method that checks whether a discourse entity should be labeled with a new variable or reuse an already established discourse variable. Instead, the parser assigns a new variable to each lexical item it encounters during the parse. This strategy leaves the task of finding and merging discourse variables to the semantic interpreter. The following two sections explain how domain specific constraints are used by the semantic interpretation process for merging the discourse variables of any lexical forms that refer to the same semantic entity, and how larger semantic structures represented by QP frames from individual sentences can be maintained in a consistent way for a paragraph-level interpretation.

6.4.1 Sentence-level semantic interpretations

Consistent discourse variables within the context of a single sentence are essential for generating a correct semantic interpretation. If an individual or a reference to the individual appears multiple times, the individual should be referred to by the same variable. If we talk about a brick that has two properties *heat* and *temperature*, the resulting Quantity frames should refer to the same brick, i.e. the frames should be constructed such that their entity slots use the same discourse variable.

On the other hand, the semantics of QP frames can be used for finding those individuals that should be referred to by the same discourse variable. Sentence (10) contains all the problems introduced in the previously discussed examples. The brick, the ground, and the temperature appear twice, but only the ground and the brick should be referred to by the same discourse variable. Moreover, the heat is only mentioned once but should be used as a quantity type in two different quantity frames.

(10) The heat flows from the brick to the ground, because the temperature of the brick is higher.

A set of forward-chaining LTRE rules is used to construct an initial semantic interpretation of the sentence. This interpretation consists of a set of QP frames plus

Intra-sentential merge (Interpretation):

1. Identify Frames and Discourse Variables that can be merged based on ruleset.
2. Suggest merge operation for candidate pairs.
3. For all candidate pairs not marked as unmergeable:
    a. For discourse variables:
        i. Create new discourse variable
        ii. Update old variable in interpretation
    b. For frames:
    - If all frame elements match:
        i. Create new frame
        ii. Update variables in interpretation
        iii. Delete old frame information
    - Otherwise mark pair as unmergeable
4. Repeat step 1 until no more merge candidates are found.

Figure 6.4: Intra-sentential merge algorithm

the remaining expressions of the general semantic interpretation produced by the parser. The QP frames use the original discourse variables from the parser output, i.e. the same individual could be referred to by two different variables.

The initial interpretation is then subjected to an intra-sentential merge procedure detailed in Figure 6.4. Similar to the rule set for constructing QP frames, another set of LTRE rules identifies discourse variables that refer to the same semantic entity. The rules produce assertions that suggest the merge of two variables or entire QP frames. The merge candidates are identified by the following two predicates:

```
(mergeVars <variable1> <variable2>)
(mergeFrames <frametype> <frame1> <frame2>)
```

The merge operation for two discourse variables leads to the creation of a new variable that replaces the two merge candidates. The merge algorithm then propagates the new variable by replacing all old instances with the new discourse variable. As a final step, the algorithm uses the predicate `mergedVars` to record the operation.

```
(mergedVar <new_var> <old_var1> <old_var2>)
```

The following rule suggests a merge operation between two variables for the entity slot of a Quantity frame, as long as the two entities belong to the same collection and have parser-assigned discourse variables.[6]

```
(rule ((:true (entity ?qf1 ?ent1)
              :var ?e1)
       (:true (entity ?qf2 ?ent2)
              :var ?e2
              :test (and (not (equal ?qf1 ?qf2))
                         (not (equal ?ent1 ?ent2))))
       (:true (isa ?ent1 ?obj)
              :var ?isa1)
       (:true (isa ?ent2 ?obj)
              :var ?isa2
      (rassert!
       (:implies (:and ?e1 ?e2 ?isa1 ?isa2)
                 (mergeVars ?ent1 ?ent2))))
```

QP frame structures use two types of frame elements: discourse variables such as `brick109`, and values such as `Hot`, `Fahrenheit`, or `Positive`. Therefore, merging

[6] The discussion of the controlled language in chapter 5 describes how user-assigned variables can be used in noun phrases to label discourse entities, e.g. for the phrase *'the brick b1'* the label *b1* would be used as the discourse variable for the brick.

two QP frames includes merging discourse variables as well as merging values. The merge operation for variable slots is essentially the same as the one for discourse variables, except that a different predicate will be used to indicate the successful merge operation.

```
(mergedVarSlot <new_expr> <old_expr1> <old_expr2>)
```

The merge operation on value slots is slightly different and will only be performed when the fillers of the two slots are 'mergeable'. Two values can be merged when they are equal or one of the values is currently unassigned. This is a conservative merge requirement that should eventually be relaxed. For example, the value and unit slots could be considered in combination to allow the use of different units.[7] A successful merge operation on a value slot will be recorded by using the predicate `mergedValueSlot`.

```
(mergedValueSlot <new_expr> <old_expr1> <old_expr2>)
```

If any of the value slots do not meet the necessary requirements for a successful merge operation, the two frames will not be merged and excluded from future merge operations. The predicate `unmergeableFrames` overrides any `mergeFrames` suggestion until an update of the value slots occurs and removes the merge restriction.[8]

```
(unmergeableFrames <frame1> <frame2> <reason>)
```

As it has been illustrated in chapter 2, capturing information about physical quantities is the central step in creating a semantic interpretation from descriptions of physical phenomena. Quantity frames. This fundamentally important role of continuous parameters is reflected by the fact that only Quantity frames are merged, when information from different sentences is joined together. Every other type of QP frame can be rebuilt based on the merged set of Quantity frames. This includes the PhysicalProcess frames, which are constructed as the final step of a sentence-level interpretation. The rebuilding operation has to be performed after any merge operation, because the modified information from two interpretations can lead to the instantiation of new QP frames.

---

[7] The length of an entity could be expressed in Centimeter in one Quantity frame, while another Quantity frame refers to the length of the same entity in inches. As long as the converted values are approximately the same, the value and unit slots could be merged. For now, we will consider these extensions as future work.

[8] The unmergeable expression will be reinstated should the update of the slot still result in incompatible filler values.

6.4.2 Paragraph-level semantic interpretations

The ability to merge frames and discourse variables allows us to produce a consistent interpretation for individual sentences. It also provides the necessary mechanisms to combine the information from several sentences to produce a multi-sentence interpretation.

Each sentence in a paragraph is parsed individually and subjected to a sentence-level (intra-sentential) semantic analysis. Once this step is complete, the information is merged with the data from the previous sentence to produce an incremental, paragraph-level (inter-sentential) interpretation. The final step consists of finding information about physical processes and the construction of PhysicalProcess frames.

Inter-sentential Merge (Interpretations A and B):

1. Combine expressions from interpretations A and B
2. Identify Frames and Discourse Variables that can be merged based on ruleset.
3. Suggest merge operations for candidate pairs
4. For all merge candidates not marked as unmergeable:
    a. For discourse variables:
        i. Create new discourse variable
        ii. Update old variable in interpretation
    b. For frames:
    - If all frame elements match:
        i. Create new frame
        ii. Update variables in interpretation
        iii. Delete old frame information
    - Otherwise:
        iv. Mark pair as unmergeable
5. Repeat step 2 until no more merge candidates can be found.
6. Rebuild frame structures from merged expressions for new interpretation
7. Perform intra-sentential merge operation on new interpretation
8. Generate process frame information
9. Store new interpretation in KB

Figure 6.5: Inter-sentential merge algorithm

The merge algorithm uses LTRE rules to find mergeable candidate frames and is run until no more merge candidates can be identified. The order in which frames are detected and merged does not matter. Figure 6.5 shows the inter-sentential merge algorithm.

The sentences in a paragraph are parsed individually, interpreted independently from each other, and stored as separate exemplars in the Interpretations knowledge base. This allows us to experiment with different sentence orders within a paragraph. Different merge sequences can be tried without the need to re-parse any of the sentences.

The two relatively simple sentences (11) and (12) each describe one particular aspect of a heat flow and an isolated property of an individual (the temperature). A merge operation combines the information about the flow processes and identify the source and destination of the flow.

(11) The heat flows from the hot brick.
(12) The heat flows to the cool ground.

There is more information that can be gathered from the merged interpretation by combining previously isolated information about properties associated with the entities. The two Quantity frames for the temperature of the brick and the ground have values associated with them and can form an OrdinalRelation frame. The symbolic values `Hot` and `Cool` are Cyc subcollections of `Temperature` and have ordering information attached to them. This data is used by the semantic interpreter to determine the ordinal relationship between the two Quantity frames, i.e. a new OrdinalRelation frame for the ‘hot brick’ and the ‘cool ground’ is instantiated in the merged interpretation.

6.5 Building process frames

The final step of the semantic interpretation process is the combination of the current frame information into a PhysicalProcess frame that takes the participants, conditions, consequences, and the current status as its frame elements.

The construction of PhysicalProcess frames start with the identification of events that are linked to physical processes. Each event in the current interpretation is checked whether it is a subcollection of `PhysicalProcess`, which is anchored in the existing ontology as a subcollection of `PhysicalEvent`. For example, the instance `flow32` is a `Translation-Flow`, which in turn is tied to `PhysicalProcess` as a subcollection. Each event that refers to a subcollection of `PhysicalProcess` creates an instance of a

PhysicalProcess frame. Information about its frame elements is gathered by running a set of process frame rules and evaluating the results by specialized procedures.

6.5.1 Process frame rules

Similar to the rules used for building general QP frame structure and merging frame information, a separate set of forward-chaining rules is used to identify information about the constituents of PhysicalProcess frames. The rules are divided into four different types, corresponding to each frame element of the PhysicalProcess frame. The following example is a rule for making a DirectInfluence become a consequence of the physical process it is associated.

```
(rule ((:true (isa ?ppframe PhysicalProcessFrame)
              :var ?ppf)
       (:true (usesQPFrame ?event ?ppframe)
              :var ?use)
       (:true (isa ?diframe DirectInfluenceFrame)
              :var ?isdi)
       (:true (constrainer ?diframe ?constrainer)
              :var ?cr)
       (:true (entity ?constrainer ?event)
              :var ?ent))
      (rassert! (:implies (:and ?ppf ?use ?isdi ?cr ?ent)
                          (consequence ?ppframe ?diframe))))
```

Process frame rules are also used for generating condition and consequence information that is not a QP frame. For example, information about the origin and destination of a transfer event, typically represented by the predicates `fromLocation` and `toLocation`, can be gathered from the QuantityTransfer frame that is associated with the event.

6.5.2 Constructing process frames

After the set of process frame rules has been run, the current reasoner data is checked for additional information about the constituents of each identified physical process.

Participants of a physical event are identified by finding all the Quantity frames that participate in the event. The entities of these Quantity Frames are treated as participants of the process frame.

Information about conditions is gathered from expressions that contain causal links between events. For example, the predicate `causesEventEvent` takes the names of two events as its arguments, with the first arguments being the condition for the

second. All expressions using such predicates with the name of the current event in their second arguments position are retrieved and analyzed. The information linked to the event in the first argument is then retrieved and treated as a condition in the current PhysicalProcess frame.

Information about consequences is gathered in a similar way used for conditions. Again, expressions containing causal information are checked. From those that contain the name of the current physical process as their first argument, the information connected to the second argument is treated as a consequence of the process.

The status information identified by the physical process rules is checked for consistency. If there is contradictory data about the status of a process, i.e. that might be active and inactive at the same time, the process status will be marked as 'undetermined'. If there is insufficient information about a process, it will be designated as 'active'. This is a fair assumption, since descriptions of physical processes generally assume that a process is active, unless the opposite is explicitly stated.[9]

After the any potential process frames have been generated, the semantic interpretation process is completed. The interpretation data is stored in a knowledge base that is separate from the Cyc knowledge base to avoid mixing user generated data from semantic interpretations with background knowledge.

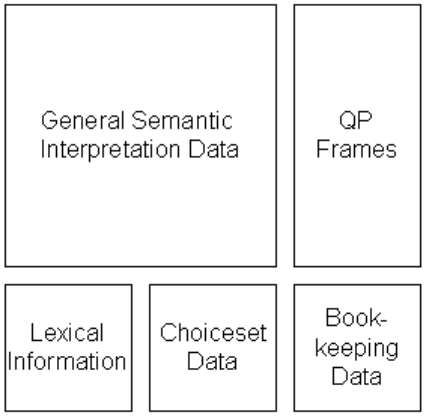

Figure 6.6: Interpretation Data

[9] The assumption that a process is active by default can easily be changed by adjusting the interpretation rules for status information.

Interpretations are stored as cases, similar to the storage and retrieval techniques used in (Forbus, Mostek, & Ferguson, 2002). The interpretation data consists of the general semantic interpretation data, the QP frame information generated by the domain-specific part of the interpretation process, as well as lexical information from the parsing process and the choice set data from the word-sense disambiguation step (Figure 6.6). Additional bookkeeping data, such as the user name, machine, and creation time, is also recorded for each semantic interpretation that stored in the Interpretations knowledge base. This information is used for indexing the sentences and their semantic interpretation data.

6.6 Summary

The system described in this chapter uses a two-step semantic interpretation process to capture information about physical processes from natural language descriptions. While the first stage is integrated into the parser and uses general knowledge provided by the background KB, the domain-specific interpretation process employs more sophisticated machinery to resolve ambiguous concepts in an evidence-based word-sense disambiguation process and builds QP frame structures by using several sets of forward-chaining rules. Separating the general semantic interpretation from the domain-specific part makes the system extensible to new domains and adaptable to different purposes such as natural language-based knowledge retrieval. These issues will be addressed in more detail in chapter 8.

Chapter 7

Examples and Evaluation

This chapter evaluates the results of our system and demonstrates the capabilities of our implementation. Using controlled language descriptions of physical phenomena as its input, the semantic interpreter described in the previous chapter produces a set of QP frame structures and expressions that include information about physical processes. The output can be evaluated by three different criteria: (1) *concept selection*, (2) *recognition of QP-specific information*, and the (3) *coverage of automatically generated process frames in comparison to hand-coded models*. Concept selection, i.e. the selection of the correct concepts for an individual word by the semantic interpretation process, allows predictions about the coverage of the background knowledge base and the ability of the word-sense disambiguation process. The recognition of QP-specific information shows the support of grammar and the coverage of the frame building rules for the construction of the appropriate frames for QP-related information in the input sentence. The coverage of the constructed process frames can be evaluated by comparing the frame information generated by the semantic interpreter against hand-coded expert models.

7.1 Word-sense disambiguation and concept selection

We have tested the evidential word-sense disambiguation module against one of sources in our corpus (Buckley, 1979) by compiling a list of lexical entries from all words in the source in all possible parts of speech. This list also included lexical entries for parts of speech that do not actually occur in the source. The lexical entries were pulled from the expanded parser lexicon. For each entry, we queried the knowledge base for corresponding concepts, i.e. concepts that match the particular part of speech of the entry. For example, the verb *house* and the noun *house* were issued as separate queries. For each query to the knowledge base we recorded one of four possible outcomes:

- *Underspecified*, if no denotational or semantic information exist in the knowledge base.
- *SemTransOnly*, if the KB contains semantic information (as expressions) for the lexical entry, but no concept corresponding to the lexical entry exists.
- *Singleton*, if exactly one concept exists for the lexical entry in the KB.

- *Choiceset*, if multiple competing concepts exist in the knowledge base

The first type of result gives us information about the coverage of the knowledge base for the entries in our lexicon. Underspecified entries are gaps in the knowledge base. Entries for which only semantic information without a single associated concept exists (SemTransOnly), no disambiguation is possible because of the missing conceptual information. The singletons do not require any disambiguation. For all choice sets, we ran the word-sense disambiguation module with all evidential tests enabled, with the following exceptions. Since the lexical entries were pulled straight out of the lexicon, there is no contextual information available, i.e. semantic information provided by other lexical entries. For this reason, the selectional restriction tests were not considered.[1] Furthermore, no user training data or user-specified terminology was used in the experiment to avoid any potential user biases. The weights for the evidential tests are listed in table 7.1.

Type of evidence	**Weight**
Quantity Type	10
Unit	10
Physical Process	10
Related to Quantity Type	5
Related to Physical Process	5
Slot Restrictor Direct match	5
Slot Restrictor Instance match	2
Slot Restrictor Subset match	1
Specialization Instance of Competitor	2
Specialization Subset of Competitor	1
Preference for POS	1
Preference in NL generation task	1
User-specified terminology	5
User preferences from interaction training	5

Table 7.1: Types of evidence and their weights

The list of words compiled from the corpus material contained 2,062 unique entries, excluding all morphological variants such as plurals and inflected verb forms. Table 7.2 shows the results for the knowledge base queries.

[1] The concepts will always match the selectional restrictions of any available semantic information associated with their own lexical entry. This is different from the use of a concept with the semantic information provided by another entry.

Result	Entries	%
Underspecified	1012	49.1
SemTransOnly	66	3.2
Singleton	598	29.0
Choiceset	386	18.7
Total	**2062**	**100**

Table 7.2: Coverage of entries

For about half of the entries, the knowledge base did not contain any semantic information or concepts. The lexicon covers more parts of speech for a particular word than the corresponding current version of the knowledge base. Moreover, the coverage of the knowledge base in terms of concepts and semantic information attached to concepts is still very selective. This knowledge engineering issue will presumably be resolved by future work (either by us or by other parties) to extend the knowledge base contents.

Parts of Speech	Number of words	%
Noun	268	69.4
Verb	91	23.6
Adjective	20	5.2
Adverb	5	1.3
Pronoun	2	0.5
Total	**386**	**100**

Table 7.3: Parts of speech for choice sets

The 386 entries that result in ambiguous information, i.e. multiple competing concepts, break down in 268 nouns, 91 verbs, 20 adjectives, 5 adverbs, and 2 pronouns (Table 7.3). The average number of competing concepts is 2.7 (Table 7.4).[2]

[2] The knowledge base contained 9 denotations associated with the noun 'sound', including the concepts `AudibleSound`, `AudioClip`, `ChannelOrStrait`, `ComputerSoundFile`, `RecordedSoundPlaying`, `Sound`, and `Sound-BodyOfWater`. The concepts `AudibleSound` and `Sound` were included both as a count and mass noun. As the data in table 7.4 shows, the large number of choices for this word is an anomaly.

Number of choices	2	3	4	5	6	7	8	9
Choice sets	230	85	49	15	6	0	0	1

Table 7.4: Number of choices per choicest

A subjective evaluation of the resolved choice sets was done to determine whether the word-sense disambiguation module selects the appropriate concepts. This evaluation was biased towards concepts relevant to the physical reasoning, i.e. if the choice for the noun 'bar' is between the concepts for 'drinking establishment' and 'unit of pressure', the concept for 'unit of pressure' was selected. The best choice was considered correct if the concept would be the subjectively best pick. It was scored as potentially correct ('maybe') if alternative interpretations would also allow one of the competitors. For example, the best choice for the word 'cycle' was a tie between `SingleRunOfADevice` and `Bicycle`. Although the corpus material uses the word only in the former sense, the latter concept is might be relevant to descriptions of physical phenomena in other sources. The selection was scored as incorrect ('wrong'), if it choice isn't likely to be used in descriptions of physical phenomena. For example, for the word 'object' the concept `Objecting-CommunicationAct` was selected as the best choice over `PartiallyTangible`.

The results of this evaluation are shown in Table 7.5. About 55 percent of the concept selections could be considered as correct, another 30 percent as potentially correct, and only 15 percent were deemed incorrect.

Evaluation result	**Number of choice sets**	**%**
Correct	215	55.7
Maybe correct	113	29.3
Incorrect	58	15.0
Total	**386**	**100**

Table 7.5: Evaluation of choice sets

Given the different levels of representational depth in the background knowledge base, i.e. the fact that abundant information exists for some areas and concepts while others are sparsely represented, the disambiguation module produced reasonable results from the current content of the knowledge base. It provides additional robustness by not ruling out information based on selectional constraints alone and allows adjustments to

the concept selection process via interactive training. Although the evidence-based approach can compensate for some unevenness in the knowledge base, underrepresented concepts with little or no semantic information are problematic and result in ambiguous or incorrect resolution results. However, the disambiguation process can be trained by resolving the choice set information for these words manually.

7.2 Recognition of QP-specific information

In the implemented system the semantic interpretation process, the controlled language, and the representational scheme of QP frames are combined to extract information about physical phenomena from natural language text descriptions. Using a number of individual sentences, this section discusses particular aspects of identifying relevant QP knowledge in the input and generating the appropriate QP frames. Each of the following examples highlights a particular aspect of the identification of information about physical processes in general and the semantic interpretation process in particular.

7.2.1 Quantities

Continuous parameters are a central concept in the QP Theory and the extraction of information about physical quantities, their values, and the direction in which they change is fundamentally important for the semantic interpretation process.

Entities and quantity types are identified by a set of interpretation rules for *attributive* (‘the hot brick’) and *possessive* (‘the temperature of the brick’) relationships, as well as *containment-* and *location-based* information (‘the pressure in the cylinder’, ‘the water at the ground’, etc.). These relationships are represented as expressions in the general semantic interpretation data. If the argument structure of the expression identifies an entity and information related to a quantity type, the appropriate quantity frame is instantiated. Each of the following examples uses this identification process for the construction of quantity frames.

(1) The brick has mass.

For example, the semantic interpreter will instantiate a quantity frame for (1) based on the possessive relationship expressed in the sentence.

```
Frame q107622 (QuantityFrame)
  Entity: brick107597
  QType:  Mass
```

7.2.1.1 Numeric value information in a quantity

Sentence (2) illustrates how values and unit can be specified as concrete numeric information. The parser identifies number and the associated unit as a measure phrase, from which the semantic interpreter will gather numeric value information for the Quantity frame.

(2) The tub contains 5 liters of water.

The quantity type does not necessarily have to be associated with the concept `PhysicalQuantity` (as it is the case with Pressure in the previous example). The semantic interpreter treats amounts of mass noun concepts as quantity types, if a Quantity frame can be constructed for this information. For example, the quantity type in the interpretation of (2) should be an amount of water in its liquid state. The semantic interpreter produces the following frame for (2).

```
Frame q108618 (QuantityFrame)
  Entity: tub108547
  QType:  (AmountFn (LiquidFn Water))
  Value:  5
  Unit:   Liter
```

7.2.1.2 Changes in physical quantities

Changes in physical quantities can be expressed directly (e.g. by the verb ‘increase’) or indirectly (e.g. as a result of a transfer event), as it has been illustrated in chapter 3. Sentence 3 contains information about a quantity type associated with an entity. This is the minimum information from which a quantity frame can be constructed. Furthermore, the sentence mentions a direction in which the resulting quantity is changing.

(3) The pressure in the cylinder is increasing.

The QRG-CE grammar allows the construction of exactly one tree for this sentence. Figure 7.1 shows the syntactic parse tree generated by the parser. Semantic and lexical information is attached to each of the tree nodes at parse time. The full node information for the main verb of the sentence is shown in the box on the right. Since this entry is a terminal node, the semantic information associated with the verb still contains placeholders such `ACTION`, `SUBJECT`, and `OBJECT` as arguments. These keywords are later replaced when phrase nodes are constructed. Semantic information from constituent terminal and phrase nodes is combined and checked for replaceable

keywords when phrase nodes are constructed. For example, the `SUBJECT` keyword is replaced with the discourse variable from the noun phrase ‘np109734’ when the verb phrase and the noun phrase are combined into the sentence-level phrase ‘slp109785’.

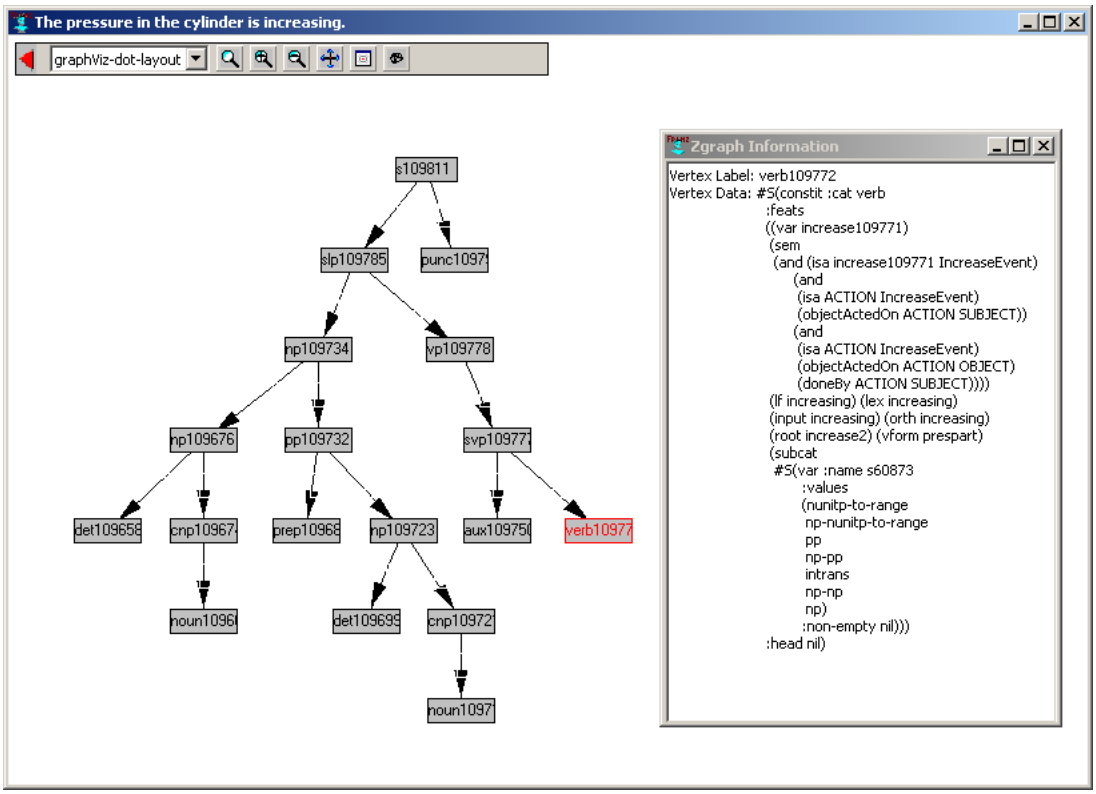

Figure 7.1: Parse tree for 'The pressure in the cylinder is increasing'

The root node of parse tree contains the general semantic interpretation data for the entire sentence. This interpretation data contains the combined semantic information from the background knowledge base, choicest data for ambiguous concepts, and lexical information about the terminal nodes. The semantic interpreter uses this data for the identification of QP-specific information and the construction of the appropriate frames based on the forward-chaining LTRE rules discussed in chapter 6.

The semantic interpretation data contains an instance of a Quantity frame, describing a positive change of pressure in a cylinder shaped object:

```
Frame q108780 (QuantityFrame)
  Entity: cylinder108684
  QType:  Pressure
  Sign:   Positive
```

Since the sentence does not contain any information about a value and unit, the corresponding optional frame elements are not present in the interpretation data either.

7.2.2 Indirect influences

Sentences that express indirect influences contain information about at least two quantities. For example, in (4) the increase in temperature has an indirect influence on the pressure in the boiler.

(4) As the temperature of the steam rises, the pressure in the boiler increases.

The sentence uses one of the indirect influence patterns discussed in chapter 3. The parser detects this pattern when sentence-level phrases are constructed and includes a `qpropEvent` expression for the two participating events (i.e. the increase in pressure and the rise in temperature) as supporting information for this pattern in the general semantic interpretation:

```
(isa increase112218 IncreaseEvent)
(isa rise112087 IncreaseEvent)
(qpropEvent rise112087 increase112218)
```

This information enables the semantic interpreter to construct an IndirectInfluence frame for the two involved quantities. The sign of the IndirectInfluence frame is determined by sign of derivative information of two Quantity frames. In this example, the two positive derivatives will result in a positive sign for the IndirectInfluence frame, and the semantic interpretation process builds the following three QP frames.

```
Frame q112444 (QuantityFrame)
  Entity: boiler112186
  QType:  Pressure
  Sign:   Positive



Frame q112445 (QuantityFrame)
  Entity: steam112073
  QType:  Temperature
  Sign:   Positive

Frame ii112446 (IndirectInfluenceFrame)
  Constrained: q112445
  Constrainer: q112444
  Sign:        Positive
```

The QRG-CE grammar contains support for the syntactic patterns, such as the THE/THE pattern, in the form of sentence-level rules. If such a pattern is detected, the grammar rule includes a `QpropEvent` expression with the appropriate arguments in the semantic interpretation data. Verb-based patterns, such as CAUSES or INFLUENCES, are covered by additional KB information attached to semTrans data for the corresponding verb entry.

7.2.3 Transfer between quantities

Although the description of a heat flow between two entities in (5) looks quite simple at first glance, it contains a lot of QP-specific information.

(5) Heat flows from the hot brick to the cool ground.

The two quantities of heat for the source and the destination of the flow and the transfer event between them will result in a pair of Quantity frames and a QuantityTransfer frame. The semantic interpretation process also recognizes the symbolic temperature values ‘hot’ and ‘cool’ associated with the entities and the underlying ordinal relationship. Furthermore, the QuantityTransfer frame identifies the roles of the brick and the ground as the source and the destination of the flow, leading to the instantiation of the appropriate DirectInfluence frames. The following set of Quantity frames is constructed by the semantic interpretation process for (5).

```
Frame q109207 (QuantityFrame)
  Entity: brick108834
  QType:  ThermalEnergy

Frame q109209 (QuantityFrame)
  Entity: flow108801
  QType:  Rate

Frame q109208 (QuantityFrame)
  Entity: ground108920
  QType:  ThermalEnergy

Frame q109203 (QuantityFrame)
  Entity: brick108834
  QType:  Temperature
  Value:  Hot

Frame q109202 (QuantityFrame)
  Entity: ground108920
  QType:  Temperature
  Value:  Cool
```

The semantic interpretation data includes two OrdinalRelation frames for the comparison of the two temperature-based Quantity frames generated from the attributive relations (‘hot brick’, ‘cool ground’). Unlike sentences in which an explicit direction of comparison is given, as in *‘The A of X is greater than the B of Y’*, the comparison between the quantities in this sentence is implicit. Both Quantity frames have matching quantity types and their entities participate in the same flow event. The semantic interpreter uses this information and the symbolic values (‘hot’, ‘cool’) to instantiate the two QrdinalRelation frames.

```
Frame or109205 (OrdinalRelationFrame)
  Quantity1: q109202
  Quantity2: q109203
  Relation:  lessThan

Frame or109204 (OrdinalRelationFrame)
  Quantity1: q109203
  Quantity2: q109202
  Relation:  greaterThan
```

The flow of heat between the two entities is captured by a QuantityTransfer frame. The information about the source and the destination of the transfer plus the Quantity frame for the flow rate leads to the instantiation of two DirectInfluence frames.

```
Frame qt109206 (QuantityTransferFrame)
  Source: q109207
  Dest:   q109208
  Rate:   q109209

Frame di109212 (DirectInfluenceFrame)
  Constrained: q109208
  Constrainer: q109209
  Sign:        Positive

Frame di109211 (DirectInfluenceFrame)
  Constrained: q109207
  Constrainer: q109209
  Sign:        Negative
```

Finally, the PhysicalProcess frame for the heat flow process contains information from the individual QP frames. Since the temperature difference between the brick and ground was not explicitly mentioned as a cause for the flow, this ordinal relationship is not included as a condition in the PhysicalProcess frame.

```
Frame physproc109214 (PhysicalProcessFrame)
  Participants:
    brick108834
    ground108920
  Consequences:
    (toLocation flow108801 ground108920)
    di109211
    (fromLocation flow108801 brick108834)
    di109212
  Status:
    Active
```

7.3 Merging frame information across sentences

The semantic interpreter allows the construction of frame information across multiple sentences. Semantic information from individual sentences can be merged into a single set of frames. The following example demonstrates the merge process by splitting up the information in (5) into two separate sentences.

(6) The heat flows from the hot brick.
(7) The heat flows to the cool ground.

The semantic interpretation process produces the following set of frames for (6). Note that the QuantityTransfer frame only fills the source and the rate frame elements, since the sentence does not contain any information about the destination of the flow. Furthermore, the OrdinalRelation frames generated for (5) are missing here, because the temperature value of the brick cannot be compared to any other quantity yet.

```
Frame q110109 (QuantityFrame)
  Entity: brick110075
  QType:  Temperature
  Value:  Hot

Frame q110111 (QuantityFrame)
  Entity: brick110075
  QType:  ThermalEnergy


Frame q110112 (QuantityFrame)
  Entity: flow110035
  QType:  Rate

Frame qt110110 (QuantityTransferFrame)
  Source: q110111
  Rate:   q110112
```

```
Frame di110114 (DirectInfluenceFrame)
  Constrained: q110111
  Constrainer: q110112
  Sign:        Negative

Frame pp110116 (PhysicalProcessFrame)
  Type:
    Translation-Flow
  Participants:
    brick110075
  Conditions:
  Consequences:
    di110114
    (fromLocation flow110035 brick110075)
  Status:
    Active
```

Figure 7.2 depicts the frame structures involved in the interpretation of (6). The isolated quantity frame on the left is the temperature of the brick. It is not integrated into the process frame, because the sentence does not specify it as a condition or consequence of the flow event.

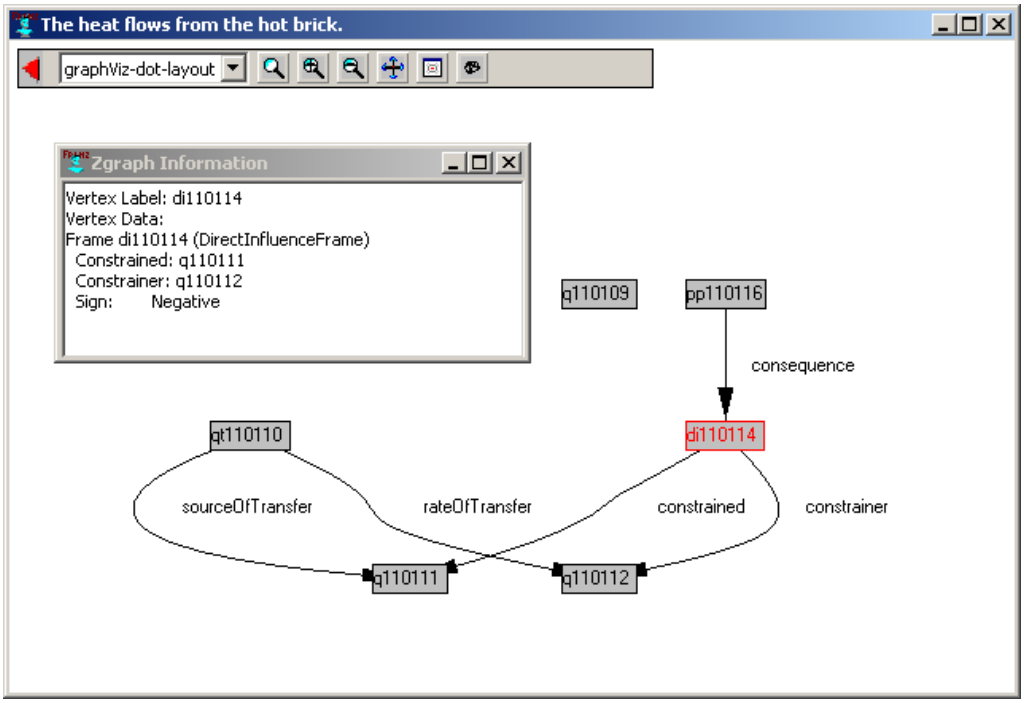

Figure 7.2: QP Frames for 'The Heat flows from the hot brick.'

The interpretation for (7) is similar to that of (6), except that the heat of the ground is identified as the destination in the QuantityTransfer frame. Figure 7.3 shows the frame structures for (7).

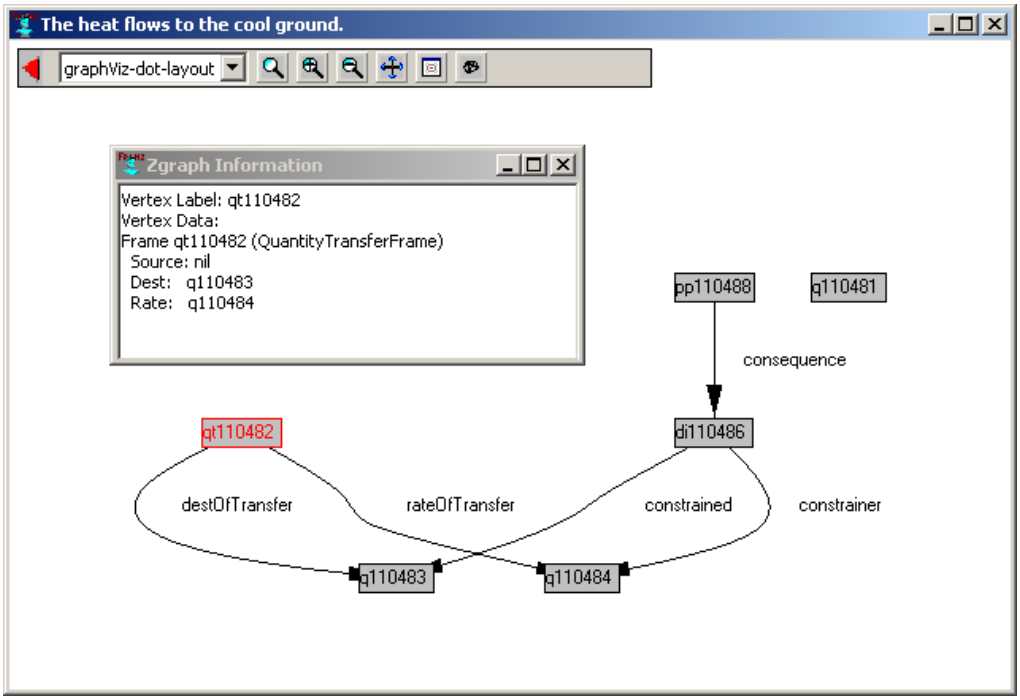

Figure 7.3: QP Frames for 'The heat flows to the cool ground.'

When the semantic interpretation data of (6) and (7) is combined, the merge algorithm identifies the two instances of a concept resulting from the noun ‘heat’ and the two instances of a flow event resulting from the verb ‘flow’ as mergeable. After the information is merged and propagated through the set of expressions, a new set of frames will be constructed. During this process, the interpreter will also detect the ordinal relationship between the two temperatures and instantiate the appropriate frames. The sum is more than its parts - merging the information from two sentences leads to the creation of new knowledge.

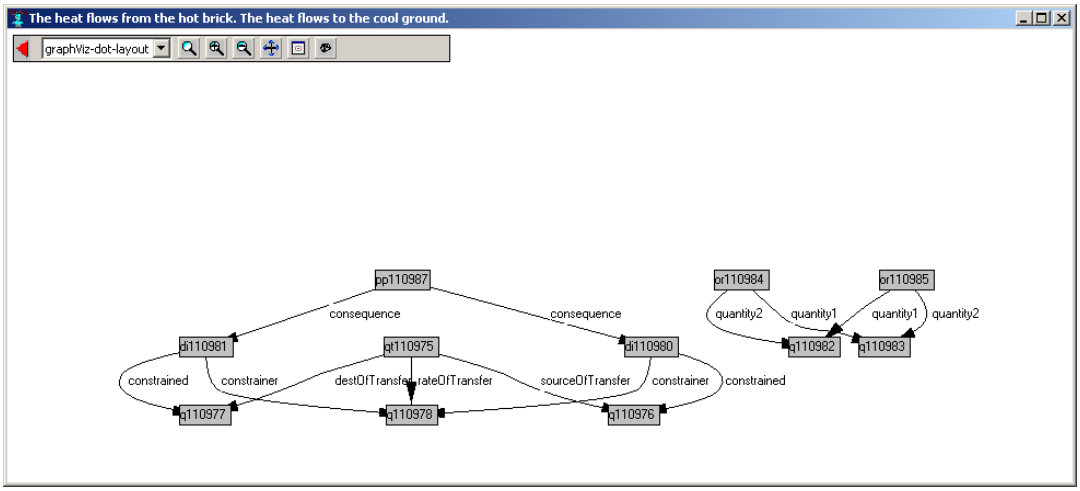

Figure 7.4: QP Frames for merged interpretations.

Figure 7.4 shows the links between the frame structures. A merge operation on the two individual semantic interpretations for (6) and (7) produces a set of frames that is semantically identical to those generated in the interpretation of (5).

7.4 Comparison against hand-coded models

The following section uses multi-sentence descriptions of two classic QP scenarios to illustrate how more complex models of physical processes can be constructed by the semantic interpreter. The resulting process frames are compared against hand-coded expert models (in the form of CML model fragments) built from the same description.

7.4.1 Fluid flow between two containers

The first three sentences (8, 9, and 10) of the description establish the scenario used for the fluid flow between two containers. Unlike the previous examples, the two cylinders are named here. The semantic interpretation process uses the labels ‘c1’ and ‘c2’ instead of creating new discourse names for each cylinder instance.

Sentence 11 describes the actual flow event between the containers. It also explicitly names the level difference as the cause for the flow. While the comparison between temperatures in the previous example was made only indirectly through symbolic value information, the two level quantities are compared directly in this example. Sentences 12 and 13 use typical syntactic patterns for indirect influences describe qualitative proportionalities.

(8) A pipe connects cylinder c1 to cylinder c2.
(9) Cylinder c1 contains 5 liters of water.
(10) Cylinder c2 contains 2 liters of water.
(11) Water flows from cylinder c1 to cylinder c2, because the level in cylinder c1 is greater than the level in cylinder c2.
(12) The higher the pressure in cylinder c1, the higher the flowrate of the water.
(13) As the amount of water in cylinder c2 increases, the pressure in cylinder c2 increases.

Figure 7.5 shows the frame structures generated by the semantic interpreter for this example.

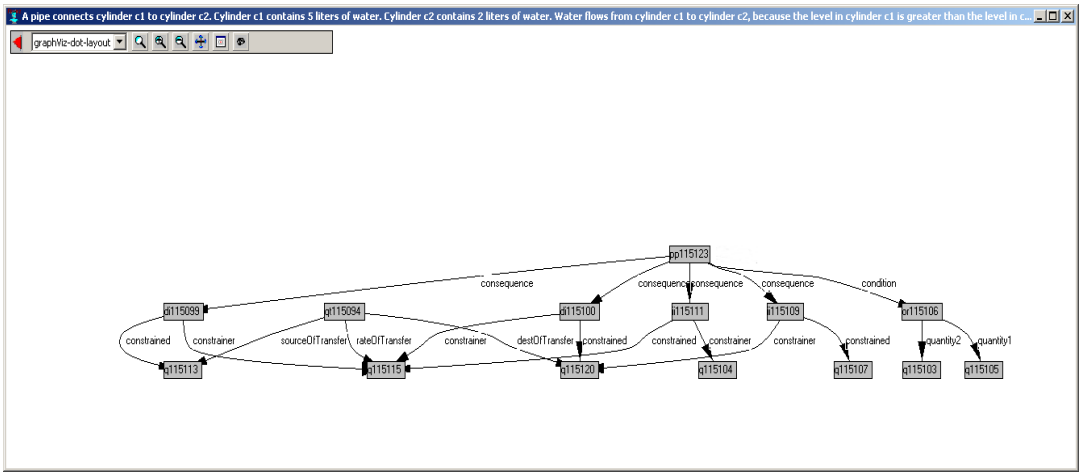

Figure 7.5: QP Frames for two-container fluid flow

The full set of QP frames produced by the semantic interpreter for the five sentences of this example is shown below. Five quantity frames are generated – for the amounts of water and the pressure in the cylinders and for the flowrate of the water (highlighted in the figure above). The labels for the cylinders are preserved throughout the merge processes.

```
Frame q115113 (QuantityFrame)
  Entity: c1
  QType:  (AmountFn (LiquidFn Water))
  Value:  5
  Unit:   Liter
```

```
Frame q115120 (QuantityFrame)
  Entity: c2
  QType:  (AmountFn (LiquidFn Water))
  Value:  2
  Unit:   Liter

Frame q115105 (QuantityFrame)
  Entity: c1
  QType:  Level

Frame q115103 (QuantityFrame)
  Entity: c2
  QType:  Level

Frame q115104 (QuantityFrame)
  Entity: c1
  QType:  Pressure
  Sign:   Positive

Frame q115107 (QuantityFrame)
  Entity: c2
  QType:  Pressure
  Sign:   Positive

Frame q115115 (QuantityFrame)
  Entity: flow115116
  QType:  Rate
  Sign:   Positive
```

A single OrdinalRelation frame is created for the different levels in C1 and C2, because the comparison in (11) is directional.

```
Frame or115106 (OrdinalRelationFrame)
  Quantity1: q115105
  Quantity2: q115103
  Relation:  greaterThan
```

The transfer of water between the two cylinders is captured by a QuantityTransfer frame, which identifies the amount of water in cylinder C1 as the source and the amount of water in C2 as the destination quantities of the flow. Since no explicit rate is mentioned yet, the semantic interpreter instantiates a default Quantity frame for the rate. The information from the QuantityTransfer frame is used to generate the appropriate DirectInfluence frames for the flow.

```
Frame qt115094 (QuantityTransferFrame)
  Source: q115113
  Dest:   q115120
  Rate:   q115115
```

```
Frame di115100 (DirectInfluenceFrame)
  Constrained: q115120
  Constrainer: q115115
  Sign:        Positive

Frame di115099 (DirectInfluenceFrame)
  Constrained: q115113
  Constrainer: q115115
  Sign:        Negative
```

The information about the qualitative proportionalities described in (12) and (13) leads to the instantiation of two IndirectInfluence frames, capturing the influence of the pressure in cylinder C1 on the flowrate of the water and the amount of water in C2 on the pressure in C2. The interpretation of (12) includes a Quantity frame for the flowrate of water, which is merged with the default rate frame instantiated by the previous sentence.

```
Frame ii115111 (IndirectInfluenceFrame)
  Constrained: q115115
  Constrainer: q115104
  Sign:        Positive

Frame ii115109 (IndirectInfluenceFrame)
  Constrained: q115107
  Constrainer: q115120
  Sign:        Positive
```

The resulting PhysicalProcess frame includes the frame for direct and indirect influences as consequences and the OrdinalRelation frame as a condition.

```
Frame pp115123 (PhysicalProcessFrame)
  Type:
    Translation-Flow
    PhysicalProcess
  Participants:
    c1
    c2
  Conditions:
    or115106
  Consequences:
    di115099
    ii115109
    (fromLocation flow115116 c1)
    di115100
    ii115111
    (toLocation flow115116 c2)
  Status:
    Active
```

A comparison of the contents of process frames with the information contained in hand-coded models is useful for the evaluation of the semantic interpretation results produced by our system. Figure 7.6 shows a CML model fragment for a water flow process and the instantiation of the two-container scenario.

```
(defModelFragment waterflow
  :subclass (flow)
  :participants ((src :type contained-stuff)
                 (dst :type contained-stuff)
                 (con :type path)
  :conditions ((connects con src dst)
               (> (pressure src) (pressure dst))
  :quantities ((flowrate :dimension rate-dimension))
  :consequences ((Qprop+ (flowrate :self) (pressure src))
                 (Qprop- (flowrate :self) (pressure dst))
                 (I- (level src) (flowrate :self))
                 (I+ (level dst) (flowrate :self))))



(defScenario two-container-example
  :individuals ((c1 :type Container)
                (c2 :type Container)
                (p  :type Pipe-GenericConduit))
  :initially ((> (level c1) 0)
              (>= (level c2) 0)
              (> (level c1) (level c2)))
  :throughout ((connects p c1 c2)))
```

Figure 7.6: Model fragment and scenario for two-container flow process

Most of the information contained in the model fragment and the scenario definition is included in the interpretation data. The PhysicalProcess frame for the water flow includes both of the cylinders as participants as well as the direct and indirect influences and the ordinal relationship between the different levels as a condition. The third participant in the model fragment, the pipe, functions as a connection between the two cylinders and is only indirectly involved in the actual flow process. The interpretation data contains assertions for this connection event. The flowrate as an internal quantity of the model fragment is also present in the information extracted from the process description, captured by the rate quantity associated with the flow event.

The interpretation data does not contain information about the two indirect influences that are symmetric to the ones found in the description, i.e. between the pressure in C2 and the flowrate, and the amount of water and the pressure in C1. Since this information has not been explicitly stated in the description, it cannot appear in an automated interpretation generated of this input.

Incomplete information is a general phenomenon found in descriptions of physical processes. Authors often try to avoid repetitions and leave out parts that are similar or symmetrical to others. By accumulating a number of different descriptions of the same processes, the information missing in individual descriptions can be filled in.[3] A more complete general model can be constructed from individual descriptions through a generalization process. We will discuss this point in more detail as future work in chapter 8.

7.4.2 Conduction heat flow – ice cube, metal rod, and coffee

Another example from *Sun Up to Sun Down* is the flow of heat from a cup of hot coffee through a metal rod, melting an ice cube frozen on the top end of the rod. This slightly bizarre scenario and the resulting heat flow could be described by the following four sentences.

(14) An icecube is on the end of a rod.
(15) A cup contains some hot coffee.
(16) The rod is placed in the coffee.
(17) Heat flows from the coffee to the cool icecube through the rod.
(18) The heat causes the icecube to melt.

The semantic interpreter produces a set of interconnected QP frames, which is shown in Figure 7.7.

[3] This assumes that there is complementary information between different descriptions, as well as sufficient overlap to make them similar to each other.

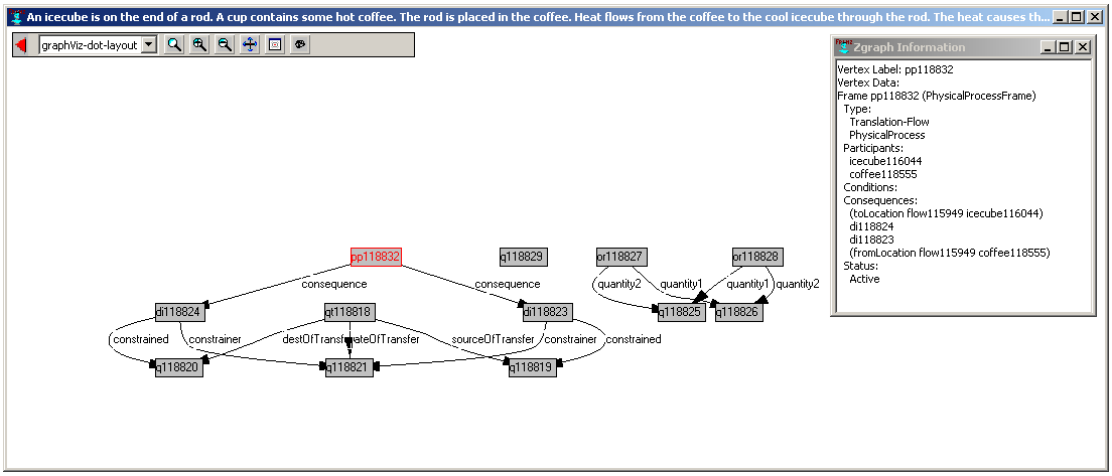

Figure 7.7: QP Frames for conduction heat flow

The temperature and the heat of the coffee and the icecube are captured by the semantic interpretation process as Quantity frames. One additional Quantity frame is generated for the amount of coffee in the cup.

```
Frame q118819 (QuantityFrame)
  Entity: coffee118555
  QType:  ThermalEnergy

Frame q118820 (QuantityFrame)
  Entity: icecube116044
  QType:  ThermalEnergy

Frame q118825 (QuantityFrame)
  Entity: coffee118555
  QType:  Temperature
  Value:  Hot

Frame q118826 (QuantityFrame)
  Entity: icecube116044
  QType:  Temperature
  Value:  Cool

Frame q118829 (QuantityFrame)
  Entity: cup115252
  QType:  (AmountFn Coffee-Beverage)

Frame q118821 (QuantityFrame)
  Entity: flow115949
  QType:  Rate
```

The Quantity frame for amount of coffee in the cup remains unconnected, because it does not participate in the conduction heat flow process. Because the temperature of the coffee in (15) and the temperature of the icecube in (17) are not compared directly, the semantic interpreter constructs OrdinalRelation frames from the symbolic values of the two Quantity frames.

```
Frame or118828 (OrdinalRelationFrame)
  Quantity1: q118825
  Quantity2: q118826
  Relation:  greaterThan

Frame or118827 (OrdinalRelationFrame)
  Quantity1: q118826
  Quantity2: q118825
  Relation:  lessThan
```

The transfer of heat between the coffee and the icecube is captured by a QuantityTransfer frame and the two DirectInfluences for the source and the destination of the flow process.

```
Frame qt118818 (QuantityTransferFrame)
  Source: q118819
  Dest:   q118820
  Rate:   q118821

Frame di118824 (DirectInfluenceFrame)
  Constrained: q118820
  Constrainer: q118821
  Sign:        Positive

Frame di118823 (DirectInfluenceFrame)
  Constrained: q118819
  Constrainer: q118821
  Sign:        Negative
```

The resulting PhysicalProcess frame includes these DirectInfluence frames as consequences. Since the temperature difference between the coffee and the ice cube is not explicitly mentioned as a condition or consequence of the heat flow process, it is not part of the PhysicalProcess frame (see Figure 7). Nevertheless, this information is part of the interpretation, because it can be used in the semantic interpretation of subsequently added sentences.

```
Frame pp118832 (PhysicalProcessFrame)
  Type:
    Translation-Flow
    PhysicalProcess
  Participants:
    icecube116044
    coffee118555
  Conditions:
  Consequences:
    (toLocation flow115949 icecube116044)
    di118824
    di118823
    (fromLocation flow115949 coffee118555)
  Status:
    Active
```

Sentence (18) does not contribute a consequence for the process frame. Although it mentions the quantity type ‘heat’, it does not refer to the actual flow event. Nevertheless, the causal relationship between the heat and the melting event is captured correctly by the semantic interpreter as a separate set of expressions.

```
(causes-Underspecified heat118817 melt118691)
(isa melt118691 Melting)
(isa melt118691 PhysicalProcess)
(isa heat118817 ThermalEnergy)
(isa icecube115129 IceCube)
(inputsDestroyed melt118691 icecube118642)
```

As in the previous example, a comparison against an expert model for this particular scenario will help to evaluate the results of the semantic interpretation process. Figure 7.8 shows the CML model fragment and scenario for the conduction heat flow between the coffee and the ice cube.

```
(defModelFragment conduction-hf
  :subclass (heatflow)
  :participants ((src :type thermal-physob)
                 (dest :type thermal-physob)
                 (path :type path))
  :conditions ((connects path src dest)
               (heat-aligned path)
               (> (temperature src) (temperature dest)))
  :quantities ((flowrate :dimension rate-dimension))
  :consequences ((Qprop+ (flowrate :self) (temperature src))
                 (Qprop- (flowrate :self) (temperature dest))
                 (I- (heat src) (flowrate :self))
                 (I+ (heat dest) (flowrate :self))))


(defScenario coffee-and-icecube
  :individuals ((coffee :type Coffee-Beverage)
                (cup :type DrinkingMug)
                (icecube :type IceCube)
                (rod :type Shaft))
  :initially ((> (temperature coffee) (temperature icecube))
  :throughout ((connects rod coffee icecube)
               (contains cup coffee)))
```

Figure 7.8: Model fragment and scenario for heat flow example

The main differences between the interpretation data and the hand-coded model fragment information concern the missing indirect influences in the PhysicalProcess frame. Analogous to the previous example, a complete description should also state the facts that the temperature difference is a condition for the heat flow and that the temperatures of the entities involved in the process influence the rate of the heat flow.

7.4.3 Other domains and types of processes

The two previous examples illustrated how models of physical processes can be constructed from multi-sentence descriptions by the semantic interpreter. Although both examples used different kinds of flows, the interpretation is of course not limited to this particular type of process. The framework of the interpretation process, the QP frame structures and the interpretation rules, is independent from the domain and type of the underlying processes.

Supporting additional domains and new types of physical processes is primarily a knowledge engineering task. Information about additional physical processes can be captured, as long as the knowledge base contains sufficient denotational and semantic information about these processes. For the motion domain we had to add (a) denotational information to tie verbs to KB concepts, (b) additional semantic information associated with KB concepts, i.e. semTrans information, and (c) link concepts with the appropriate collections for physical processes and quantity types. The first two tasks would benefit from an integration of the FrameNet data with the contents of the Cyc knowledge base. For example, frame element information could be used to generate template expressions for semantic information about participants and props. The following two sections take a closer look at knowledge engineering issues and the integration of linguistic and ontological resources.

7.5 Rewriting and interpretation issues

As the conduction heat flow example illustrates, missing information in the description and incomplete background knowledge are the two main causes for incomplete models. Writers of textbooks and popular science literature often assume some familiarity with basic world knowledge. The author can therefore leave out some parts of the descriptions and expect the reader to 'fill in the blanks.' A similar assumption cannot be made when the text of a single description is processed by an automated system. To avoid these problems, descriptions in QRG-CE have to be more elaborate than their unmodified counterparts. Alternatively, individual descriptions of the same process type could be accumulated to abstract a more complete general model of the underlying process (see chapter 8).

The examples presented in this chapter were chosen to illustrate the most interesting features and capabilities of our implemented systems. However, the system is not limited to just a few hand-written examples. To analyze the potential limitations and problems did we encounter when rewriting material describing physical phenomena, we have selected ten paragraphs from different text sources. The corpus material covers a variety of different skill levels and ranged from a children's book on weather (Lehr, Burnett, & Zim, 1987) to university textbooks on meteorology (Moran & Morgan, 1994) and naval engineering (Gritzen, 1980). Appendix 2 contains the original paragraphs and their controlled language counterparts.

The average number of sentences for each paragraph doubled during the rewriting process, with an average length of 5.9 sentences per paragraph in the source text and 11.3 sentences in the rewritten versions. This is mostly due to the fact that the source material contains longer sentences (15.7 words per sentence on average) compared to the rewritten, controlled language version (8.9 words per sentence on average).

However, the average number of words per paragraph grows only slightly from 92.4 to 100.2 words.

Some grammatical limitations of the controlled language make the rewriting process slightly complicated. Among them are the missing support for coordinated conjunctions, as in *'the water and the oil are flowing though the pipe'*, compound nouns ('water vapor'), passive constructs, such as '*the ball is placed in the box*', and the support for different verb tenses, temporal ordering ('after'), and measures ('daily', 'per cent'). This kind of information contained in the original is currently lost in the rewritten text. These limitations are at the focus of future extensions to the system.

Difficulties are also caused by the fact that proper nouns and terminology need to be defined in the lexicon before the parse is attempted. If a proper noun is not defined, it will either be treated as a label, if it appears together with a common noun in a noun phrase, or as an unknown word, which will most likely prohibit the construction of a complete parse tree for the current sentence. The former outcome could be used as an interesting workaround for the requirement of prior definitions. The proper noun can be used as variable in conjunction with a common noun, e.g. 'the man Joe' instead of just 'Joe'. In this case, the interpretation process will treat 'Joe' as an instance of the concept `man` and associate all the relevant semantic information from the knowledge base with it, i.e. `(isa Joe AdultMalePerson)`. Special cases of undefined but frequently encountered words are compounds such as 'relative humidity' or 'heat engine'. These terms can be defined in the lexicon as hyphenated entries, such as 'heat-engine' or 'relative-humidity'.

To find out how extensive the problem of undefined proper nouns and the lack of domain-specific vocabulary is, we have analyzed a representative part of the corpus material for words not covered by our lexicon. The *Sun Up to Sun Down* part of our corpus contained 93 missing words (out of a total of 3,319 words, or 2.8%) that were not part of the COMLEX 3.1 data used for the parser lexicon. More than half of these words (53, or 56.38%) were hyphenated compounds, such as 'house-heating' or 'water-filled'. The remaining missing entries were mostly place names and adjectives such 'Australia' or 'Irish', as well as technical terms such as 'absorber' or 'biomass'.

While missing lexical entries manifest themselves primarily in incomplete parses, a major limitation that often prevents a successful semantic analysis of the input text is the mapping between the parser lexicon and the Cyc lexicon. This is not surprising, since the Cyc lexicon is smaller than lexicon used by the parser. While the parser lexicon contains 86,297 expanded entries based on 39,533 unique entries in the COMLEX data, the Cyc lexicon defines merely 16,552 instances of the Cyc collection

`EnglishWord`.[4] Even if the same word is defined in both lexicons, orthographic differences can still prevent a successful mapping.

Another common source of problems are unconnected lexical entries and undefined concepts in the knowledge base. Lexicon entries are unconnected if some lexical information is missing that would be required for finding the appropriate concept, such as missing part of speech data or denotational information. We have encountered some instances in which a Cyc lexicon entry and an appropriate concept existed in the knowledge base but were not linked by a denotation. In other cases, part of speech information was omitted for a word sense, preventing a successful lookup of semantic information for a word.

For some lexicon entries the knowledge base does not contain any defined concept at all, i.e. no denotational information is associated with a particular word in the lexicon. The result is the same as for unconnected lexical entries – no concept or semantic information can be retrieved from the knowledge base.

Underdefined concepts are a less problematic case. Even if a concept can be retrieved for a particular lexical entry, we have often found no semantic information attached to in the form of semTrans expressions. For nouns, adjectives, and adverbs this usually is not a real problem, as long as the concept is tied correctly into the ontology. However, for verbs and prepositions the additional semantic information is important, since it ties different pieces of information within a sentence together during the construction of phrase nodes when keywords in semTrans expressions are replaced by discourse variables. This leads directly to another set of problems. In a few cases, the semantic information in the knowledge base showed inconsistencies, such as reversed argument structures, wrong frame keywords, and incorrect part of speech information. These inconsistencies are rare and easy to correct, but they are difficult to detect in advance.

Besides problems with the semantic information retrieved from the knowledge base, the semantic interpretation can produce incorrect results originating from problems in the frame building process. Since the QRG-CE grammar provides support for only the most frequent syntactic patterns (chapter 3), some new, infrequent pattern might not be detected and parser fails to include the necessary support information for the semantic interpreter. However, a controlled language does not have to support every possible pattern and we can restrict the use of these constructs to just the defined set. Furthermore, the interpretation process can fail when the general semantic interpretation data contains expressions that are not recognized by the frame building

[4] Based on our subset of the Cyc knowledge base, version 576, October 2003

rules as relevant for the instantiation of a particular QP frame. In such cases, new rules must be added, assuming the input text is support by QRG-CE.

7.6 Integration of linguistic and ontological resources

Over the course of this project, it became clear that the semantic interpretation process would greatly benefit from the integration of the FrameNet data into the Cyc knowledge base. The current way of specifying semTrans information for concepts is not very sophisticated, semantically undifferentiated, and too dependent on lexical information.

For example, the semantic information attached to verbs does distinguish between different grammatical forms of usage, such as transitive and intransitive use. The following `verbSemTrans` expressions show the three different pieces of semantic information for the verb 'move'.

```
(verbSemTrans Move-TheWord 0 IntransitiveVerbFrame
  (and (isa :ACTION MovementEvent)
       (primaryObjectMoving :ACTION :SUBJECT)))

(verbSemTrans Move-TheWord 1 IntransitiveVerbFrame
  (and (isa :ACTION ChangeOfResidence)
       (performedBy :ACTION :SUBJECT)))

(verbSemTrans Move-TheWord 2 TransitiveNPCompFrame
  (and (isa :ACTION CausingAnotherObjectsTranslationalMotion)
       (objectActedOn :ACTION :OBJECT)
       (doneBy :ACTION :SUBJECT)))
```

The intransitive verb form distinguishes between two word senses, i.e. `MovementEvent` as in "The car is moving" and `ChangeOfResidence` as in "I moved [away]." The transitive verb form captures only one verb sense, i.e. `CausingAnotherObjectsTranslationalMotion` as in "I moved the vase." Compared to the Motion frame discussed in chapter 4 and its nine frame elements (`area`, `carrier`, `distance`, `duration`, `goal`, `path`, `source`, `speed` `theme`), the `semTrans` information found in Cyc is quite coarse and offers only two role relations, `objectedActionOn` and `doneBy`.[5] The FrameNet data contains a much more fine-grained structure of frame elements and their usage in certain combinations as exemplified by lemmas.

---

[5] The Cyc KB has `relationIndicators` tied to the lexical entry of the verb 'move' that indicate that certain relations might or might not hold. However, this information is just a hint towards a possible relation and cannot be used in the same way as the semTrans expressions.

An integration of FrameNet with the Cyc knowledge base would be beneficial for both. It gives Cyc access to fine-grained semantic information about verbs and noun. In return, the FrameNet data could be tied into the ontological structure of the knowledge base and allows it to be used in reasoning systems. The frame layer would provide a structure for the `semTrans` information and allows the selection of the right set of semantic information based on the frame elements in a particular instantiation of a frame. The fact that the representation language of Cyc has its roots in frame-based systems, as evidenced by the notion of slots and units in (Lenat & Guha, 1989), should facilitate this integration.

There are at least two different ways to combine Cyc and FrameNet. The first approach would use frames as an intermediate link between the lexical and the actual KB concepts, isolating the conceptual information from the lexicon through the frames layer. Alternatively, expressions could link the lexical information to frames and KB concepts at the same time, preserving the existing denotational information. While this approach is less intrusive, the first solution might be cleaner because it prevents 'shortcuts' via the old denotation expressions. Both solutions follows the model suggested in this thesis by using frames an intermediate representational layer between natural language and abstracted knowledge such as KB concepts. The use QP frames as an intermediate representational layer between the natural language input and the final representations in the form of KB expressions are an important step towards this integration. As it has been described in detail in chapter 4, QP frames are designed as a specialized extension to FrameNet. The compatibility between the two representations is an important aspect for a future integration.

7.7 Summary

The semantics of QP Theory play an important role in capturing information from natural language descriptions of physical phenomena. As illustrated by the examples in this chapter, the sets of frames generated from multi-sentence descriptions of physical phenomena are comparable to the information found in manually constructed models of physical processes. However, we have encountered some limitations of the interpretation process that are connected to (a) the input material, i.e. the controlled language descriptions and (b) the background knowledge, i.e. the lexical and semantic information contained in the Cyc KB and the parser lexicon.

The quality of the input descriptions plays a crucial role in interpretation process. As long as everything is properly described in terms of the controlled language, including the use of typical syntactic patterns for the constituents of QP Theory, the semantic interpreter is able to construct the appropriate QP frames. The problem is that authors

of textbooks usually assume some degree of world knowledge and leave of ‘obvious’ parts. As a consequence for writing descriptions of physical processes that can be interpreted successfully, implicit world knowledge has to be stated explicitly. The semantic interpreter cannot ‘read between the lines’ to fill in omitted world knowledge that a human reader might have. For example, a human reader knows placing a kettle on top of a stove results in a thermal connection between the two objects and allows a conduction heat flow to occur. Furthermore, causal connections outside QP semantics have to be made explicit, e.g. if a heat flow process is caused by the temperature difference between two objects, the author has to state this fact explicitly. Otherwise, the ordinal relation will not be included as a condition for the process, as illustrated by the merge example in section 7.4.

Knowledge base issues can be divided in four categories: missing lexical entries, missing and underrepresented concepts, missing or incomplete semantic information, and ordinary knowledge engineering bugs. Many of these problems will disappear over time, as the contents of the knowledge base grow. Some of the limitations can also be resolved by an integration of additional linguistic and ontological resources, such as WordNet and FrameNet, with the existing background knowledge. However, more background knowledge also means a greater amount of conceptual information that needs to be disambiguated.

Chapter 8

Conclusions

In this thesis we have shown that Qualitative Process Theory, an established formalism for expressing mental models of physical phenomena, can be an essential component of natural language semantics. Understanding the connections between the ideas of QP Theory and their manifestation in natural language descriptions sheds light on how knowledge about physical phenomena is communicated. We started the investigation with a corpus analysis of the syntactic forms in which information about physical quantities and constituents of QP Theory appear in natural language. Physical quantities are a fundamental element of Qualitative Physics and provide the basic building blocks for the interpretation of descriptions of physical processes. Chapter 2 showed how the information about the five constituents of physical quantities can be identified in natural language. Chapter 3 extended the analysis to higher-level constituents of QP Theory and show that distinct patterns can be found for three of these constituents. The results of the analysis allowed us to derive grammar and interpretation rules for QP-relevant knowledge.

Information about physical processes is captured in an intermediate representational layer that links the natural language input with QP semantics. Inspired by frame semantics and intended as an extension to FrameNet, we have recast QP Theory as a set of specialized frame structures. The frames are formally identical to QP Theory and allow the application of standard qualitative reasoning techniques on these representations.

Based on the analysis of the syntactic realizations of QP constituents in natural language, we have designed a controlled language for describing physical phenomena in a readable, yet less ambiguous subset of English. The language encodes these realizations as grammatical rules and supports the semantic interpretation process by reducing ambiguity. The controlled language is implemented as a context-free grammar for a bottom-up parser.

Our controlled language does not restrict the number of word senses for lexical entries. The parser retrieves semantic information from a background knowledge base and constructs a general, often ambiguous, interpretation. The background knowledge base consists of a subset of the Cyc knowledge base as well as other sources. A word-

sense disambiguation module is used to find the most appropriate semantic data associated with a lexical entry. The disambiguation algorithm collects and weighs various types of evidence supporting alternate word senses. In addition to handling the naturally occurring ambiguity in language, it also helps overcome inconsistencies in the knowledge base, such as missing lexical entries, non-aligned argument structures and erroneous part of speech information. Finally, the semantic interpretation process constructs QP frame structures from the disambiguated semantic data via sets of forward-chaining rules. Information from multiple sentences is merged to generate a paragraph-level semantic interpretation.

The output of the system has been evaluated by three criteria: (1) *concept selection*, (2) *recognition of QP-specific information*, and (3) *coverage of automatically generated process frames in comparison to hand-coded models*. On an exhaustive list of words from a representative part of our corpus the word-sense disambiguation process selected the correct concepts for more than 55% of the ambiguous words. Another 30% were potentially correct when domain specific constraints are applied. The recognition of QP-specific information is demonstrated by the ability of the controlled grammar and the semantic interpretation rules to identify QP-related information and to construct the appropriate frames for the input. As illustrated by a number of examples in chapter 7, the grammar and the semantic interpreter recognize all of the QP-specific constructs identified by the corpus analysis. The frame information constructed by the semantic interpreter closely matches hand-generated expert models, as long as the natural language descriptions contains all the relevant details. However, authors frequently assume that their readers possess a certain degree of world knowledge and leave out ‘obvious’ information from their descriptions. Such facts have to be stated explicitly in a description that can be processed by our system. The semantic interpretation process can also be hampered by erroneous information in the background knowledge base, such as missing and underrepresented concepts, non-existent lexical entries, and incomplete semantic information for concepts. These are primarily knowledge engineering issues, which will be addressed in the future work section of this chapter.

8.1 Related work

Extracting information from coherent pieces of text such as simple stories or newspaper articles has been one of the big challenges in Artificial Intelligence and Computational Linguistics for almost as long as these fields have existed. Our work primarily follows the research in deep semantic text understanding that analyzes and interprets sentence structures, rather than just skimming the input text and extracting pieces of information.

8.1.1 Text understanding and Information extraction

Early text understanding programs came out of efforts in machine translation and natural language dialog systems (Charniak, 1972; Winograd, 1972; Woods, Kaplan, & Nash-Webber, 1972). With the advent of frame-based representations (Fillmore, 1968, 1976; Minsky, 1975) and semantics-oriented theories such as Conceptual Dependency Theory (Schank, 1975; Schank & Tesler, 1969) research in text understanding made major progress. SAM (Cullingford, 1978) used predefined scripts to understand simple, stereotypical stories and newswire articles, but most of its actual knowledge was already encoded in the script itself. While SAM analyzed its input in depth and generated a potentially large number of script-based inferences, FRUMP (DeJong, 1979, 1982) used a different approach and processed news stories at a much shallower level, extracting just the gist of each story. BORIS (Lehnert et al., 1983) was another attempt at in-depth story understanding, which unfortunately required a lot of hand-coded knowledge and worked well for just a few short examples. These early attempts at deep semantics identified several key problems in natural language understanding: syntactic and semantic ambiguity, frame selection, discourse processing, and the importance of background knowledge.

As a reaction to the problems with deep semantic processing and as a response to the fact that more and more information became available online, information extraction techniques using only shallow semantics were developed. The goal of information extraction is not to fully analyze und 'understand' a text sentence by sentence but to extract only the information one is interested in for a specific task. The use of specialized syntactic patterns for capturing QP-related information (chapter 3) relates our work to some of the ideas found in the information extraction literature. For an overview of information extraction and retrieval techniques see (Pazienza, 1997, 1999), in particular (Grishman, 1997) and (Wilks, 1997).

The DARPA-initiated TIPSTER program focused information extraction research at government, industrial and academic research institutions with the goal to provide the intelligence community with improved operational tools for processing extensive text sources (Grishman & Sundheim, 1996; Voorhees, 1999). The TIPSTER program picked up ideas of the earlier script-based natural language understanding systems such as FRUMP. However, the focus of the program was on primarily skimming a large number of documents and extracting relevant information based on predefined templates, not on deep semantic text understanding as in the earlier systems.

TIPSTER spawned a number of important information extraction projects, including the development of systems based on mark-up languages such as the Alembic Workbench (Day et al., 1997) The Alembic Workbench is a useful tool for 'tagging' parts of the input text - by hand or by using a set of rules - for a subsequent extraction step. Related to this development is the DeepRead reading comprehension system

(Hirschman, Light, Breck, & Burger, 1999) which uses the Alembic name tagger (Vilain & Day, 1996). Despite its name, DeepRead does not attempt a deep semantic analysis of its input material and instead uses simple bag-of-word techniques to identify sentences that contain information relevant to WH-type comprehension questions. Early in the course of our research we have experimented with the Alembic Workbench to identify information about physical phenomena in our corpus material, but the process of marking all the information related to constituents of physical processes (instead of just named entities or locations based on a template) in unrestricted text by hand turned out to be very tedious and time-consuming. We abandoned this approach in favor of using a syntactic parser and developed the restricted input language described in chapter 5. Other IE systems for that participated in the TIPSTER text program include FASTUS (Hobbs et al., 1996) and PROTEUS (Yangarber & Grishman, 1997). Another project, BBN's IdentiFinder (Bikel, Miller, & Weischedel, 1997; Bikel, Schwartz, & Weischedel, 1999) has already made the transition into a commercial text-retrieval product.

The PROTEUS project uses an interesting algorithm for discovering new patterns of knowledge from unannotated text. Based on a small set of seed patterns, the ExDisco algorithm (Yangarber & Grishman, 2000b) first partitions the corpus into relevant and non-relevant documents. It generates a number of candidate patterns and ranks them by relevance from the set of relevant documents, i.e. those in which at least one instance of a seed pattern was found. The highest relevance is assigned to those patterns that appear most frequently in the set of relevant documents and the least often in the non-relevant document set. The highest-ranking candidate patterns are then added to the set of seed patterns and used for another iteration. Each generation of added pattern will contribute to a lesser degree to a confidence score associated with the relevance of a particular document, i.e. the original seed patterns contribute the most to this score, while the latest generation of added candidate pattern contributes the least. The iteration stops after a limit is reached or no more patterns could be found and added.

AutoSLOG (Riloff, 1993) and its successor AutoSLOG-TS (Riloff, 1996) also use pattern discovery techniques. While its original version required an annotated training corpus, AutoSLOG-TS got around this limitation by using statistical feedback. It needed only a pre-classified training corpus, i.e. sorted into relevant and non-relevant documents. Depending on the corpus, sorting the documents is more effort than defining a small set of initial pattern as in ExDisco. Text-classification algorithms such as (Scott & Matwin, 1998) would provide a possible solution for this problem. Other systems for automated pattern discovery include CRYSTAL (Soderland, Fisher, Aseltine, & Lehnert, 1995) and WHISK (Soderland, 1999). RAPIER (Califf, 1998; Califf & Mooney, 1998) is a symbolic system that uses a part-of-speech tagger (Brill,

1992, 1994) and WordNet (Fellbaum, 1998) to generate ELIZA-style rules with filler patterns. (Soderland, 1999) provides a comprehensive comparison of these systems.

Although the primary use of automated pattern discovery techniques is in information extraction applications, this research is relevant for the work in this dissertation. The patterns presented in chapter 3 were extracted by hand from a small corpus. Automating the discovery of patterns for information about physical processes and including them as rules for the semantic interpretation process would benefit our existing system.

The idea of using patterns for knowledge extraction has also been suggested for deep semantic processing. Clark's knowledge patterns approach treats natural language understanding as 'scene building' from small components of knowledge (Clark & Porter, 1997; Clark, Thompson, & Porter, 2000). The input text provides a path along which background knowledge is pulled in. The result will be a larger description, consisting mainly of background knowledge clustered around the concepts extracted from the input text. A similar idea can be found in (Staab, Erdmann, & Maedche, 2000) which ties a web-based knowledge representation language to re-useable patterns, with the goal of identifying patterns that allow a translation between different representations. Each pattern includes a particular structure that contains the core elements of a pattern and templates for examples.

Information extraction systems are used to skim a large number of documents for task-specific pieces of relevant data from text. Since they can only employ shallow semantic models for efficiency reasons, the capabilities of these systems are limited when an in-depth analysis of the source text is required. The TANKA project (Barker, Delisle, & Szpakowicz, 1998) is an attempt at using full-text parsing and deep semantics in natural language understanding for building semantic representations from technical text (Copeck, Barker, Delisle, Szpakowicz, & Delannoy, 1997).

TANKA consists of two major components: the DIPETT parser (Delisle & Szpakowicz, 1995) and the HAIKU semantic interpretation module (Barker, 1998). DIPETT is a syntactic parser that tries to construct complete parse trees for individual sentences instead of just parsing for certain semantic patterns. Similar to the approach used in the development of the QRG-CE grammar, DIPETT does not make use of any particular grammatical theory but employs a number of rules based on (Quirk, 1985). The parser returns only a single parse tree for the best interpretation of a sentence and allows the user to rearrange the tree. DIPETT builds its lexicon from scratch by using a part-of-speech tagger on the corpus material. This approach results in a smaller lexicon, tailored towards a particular corpus. The HAIKU semantic interpreter is used to identify the semantic relationships within the syntactic information supplied by the DIPETT parser. It includes modules for three different tasks: a noun modifier analysis

module (Barker, 1997), a case analysis module for relationships between a verb and its arguments (Barker, 1996), and a module for analyzing relationships between connected clauses (Barker, 1994; Barker & Szpakowicz, 1995).

HAIKU uses only a minimum of hand-coded initial semantic knowledge and relies on an interactive process to resolve ambiguities. The bracketing algorithm of HAIKU's noun modifier component is a possible solution for dealing with the compound noun problem mentioned in chapter 5. The case analysis module is of interest to our work, since it is also inspired by Fillmore's notion of case frames. It constitutes an alternative approach to the use of the FrameNet data, which was not available at the time HAIKU was built.

The KANT system (Nyberg & Mitamura, 1992) is an example of the successful deployment of a controlled language system in a large-scale commercial environment. It has been deployed as an application used by Caterpillar for the creation of documentation for heavy machinery (Nyberg, Kamprath, & Mitamura, 1998). KANT is a knowledge based machine translation project with an interesting modular architecture. It uses an intermediate representational layer that is independent from the source and target languages. The languages themselves have an explicitly coded lexicon, specialized grammars, and semantic rules that operate on the internal structures of an interlingua layer. The interlingua descriptions uncouple the source and target languages and provides the semantic 'glue' between them.

KANT uses a controlled language for creating the input documents and supports this process with a collection of authoring tools for rewriting and validating sentences in the controlled language (Mitamura & Nyberg, 2001). Although the lexicon for the controlled language initially required an unambiguous mapping between words and semantic concepts in the knowledge base, this restriction proved difficult under realistic conditions (Nyberg et al., 1997). Later versions of the system took a more pragmatic approach and allowed multiple meanings per word, similar to the design decisions we made in the development of QRG-English (chapter 5). To resolve lexical and semantic ambiguities KANT uses an interactive process during the creation of the document. Furthermore, tight grammar rules and domain knowledge help to avoid syntactic ambiguities. The acquisition of new knowledge base content is aided by editors for the domain model and semi-automated tools for building syntactic lexical and sets of interpretation rules. The approach described in (Nyberg et al., 2002) is similar to our work, as it uses the KANT system to capture semantic knowledge from simple descriptive texts. The interlingua representations for sentences written in KANT Controlled English are merged together for a paragraph-level interpretation by finding interlingua concepts with overlapping slots and compatible slot values.

8.1.2 Acquisition of lexical and conceptual knowledge

The acquisition of new concepts and relations between concepts from natural language text plays in important role in adding new information to the background knowledge base. Defining new concepts and relations by hand is a tedious and expensive process. Instead, information could be extracted from a document to build up new knowledge from already existing information in a bootstrapping process.

Knowledge acquisition systems can be classified as those that operate primarily without any user intervention, e.g. (Hahn & Schnattinger, 1998; Mooney, 1987; Schnattinger & Hahn, 1997; Wiemer-Hastings, Graesser, & Wiemer-Hastings, 1998) and those that are supervised, e.g. (Bareiss, 1989; Knight, 1996). Not surprisingly, supervised learning tends to lead to better and more focused results, i.e. the learned concepts and the relations between them are more likely to be relevant knowledge for the selected task and more similar to the information a knowledge engineer would have added. However, the amount of human interaction with the system to sort out irrelevant information and to resolve ambiguities must not be neglected and might pose a major problem in the knowledge acquisition process. Unsupervised concept learning avoids these difficulties at the risk of adding non-relevant information to the knowledge base. Filtering out the non-relevant information might again require human interaction.

In addition to a background knowledge base containing an ontology of existing concepts, most natural language processing systems also use a collection of particular instantiations of conceptual knowledge. For example, (Clark & Matwin, 1992) describe a system that uses two representational layers. The first one only includes the abstract terms of background knowledge, while the second layer provides connections between the background knowledge terms and facts from a library of examples. Their model assumes that the background knowledge provides a set of plausible rules for making these connections, as well as a set of plausible definitions for the terms in those rules. They also assume an existing quality metric for evaluating the connection. For predicting the value of certain parameters in an economic model based on numeric data sets, an already existing qualitative economic model is assumed as background knowledge. In (Clark & Matwin, 1993) new labels `I*+` and `I*-` are introduced to denote self-stabilizing feedback loops: `I*+(X,Y)` means that if X is increased, then initially Y will rise; eventually the rate of increase `dY/dt` will fall until Y reaches a constant value. For short time scales `I*+` behaves like `I+`, for long time scales `I*+` is like `Qprop+`. This model even works for incomplete background knowledge, as demonstrated in (Matwin & Rouget, 1996).

The text understander described in (Hahn & Schnattinger, 1998) uses a model of automatic acquisition of new concepts from natural language text by applying a

'quality-based' approach via rules that provide evidence and support for certain concept hypotheses. The system generates hypotheses for unknown concepts based on the grammatical information provided by the parser and the conceptual interpretation of the parse tree. The hypotheses are weighted along different ‘quality dimensions’ depending on the type of evidence that led to their generation (Schnattinger & Hahn, 1998). For example, given a particular device referenced by a proper noun, the text understander can suggest potential functions or roles of device based on the contextual parse data and conceptual information provided by a background knowledge base. This approach is similar to the use of evidential reasoning techniques in (Everett, 1999). (Cardie, 1994) presents a general framework for the acquisition of domain-specific knowledge, using case-based reasoning techniques to resolve lexical, syntactic, and semantic ambiguities. The case base of ambiguity resolution episodes, created in a supervised setting, is applied to ambiguities in novel sentences.

In addition to learning and disambiguating conceptual knowledge, it is also important to extend the existing lexicon by acquiring new lexical entries. No lexicon is complete for every possible application. In technical domains, the lexicon usually contains a large number of specialized terms, each of them tied to particular domain-specific concepts. Instead of adjusting the lexicon to a new corpus by hand, the information produced by the parser about the context in which an unknown word is found can be used to produce new lexical entries. This strategy works particularly well for feature-rich grammar systems such as HPSG (Pollard & Sag, 1994). If a word occurs multiple times certain features inserted as defaults can be eliminated to make the lexicon entry more specific (Erbach, 1990; Kilbury, Naerger, & Renz, 1992). (Barg & Walther, 1998) treats unknown information as preliminary data that can be revised by a generalization or specialization process. It even allows a modification of existing lexical entries, based on the information supplied by the parser.

8.1.3 Ontologies and knowledge bases

In addition to systems that learn new lexical and conceptual knowledge for a particular domain or a defined set of tasks, a number of research projects focus on capturing general world knowledge in ontologies to support common-sense reasoning in addition to or based on particular domain knowledge. Early attempts had a narrow focus on particular domains (for example, Hayes's ontology for liquids (Hayes, 1985)), while more recent developments tend to specialize on the upper ontological divisions. This is most evident in the heavily debated IEEE effort for a standardized upper ontology (Niles & Pease, 2001a). One of the proposed candidates for the IEEE standard is SUMO (Niles & Pease, 2001b; Pease, Niles, & Li, 2002). A comparative review of ten different ontologies, including Cyc and WordNet, can be found in (Noy & Hafner, 1997). While the main aspect of this dissertation is not ontology development, parts of this work required several additions to the background

knowledge base, such as the definitions of the QP Frame structures in chapter 4 and extensions for capturing QP-specific information during the semantic interpretation process in chapter 6.

Widely used knowledge bases such as Cyc, WordNet, or FrameNet have shown that it takes a massive effort to capture sufficient knowledge to support natural language processing and commonsense reasoning (Fellbaum, 1998; Fillmore et al., 2001; Guha & Lenat, 1990). These systems demonstrate that there is no silver bullet, no small set of primitives, facts, and rules, if the goal is to move beyond small, domain-specific systems.

Ontologies have been used in natural language processing by a number of systems such as (Bateman, 1993; Burns & Davis, 1999; Dahlgren, 1988; Mahesh & Nirenburg, 1995). The MikroKosmos ontology supplies conceptual knowledge for lexical representations and to provide constraints for the semantic interpretation process. Although the MikroKosmos ontology covers a larger number of concepts than earlier domain-specific ontologies (Hayes, 1985; Mars, 1993), the contents of the Cyc knowledge base are by far more general. The NLP-specific contents of the Cyc knowledge base have also been a central part of the KRAKEN system (Panton et al., 2002). Dahlgren's focus on commonsense knowledge as a guiding principle for natural language processing is very similar to the ideas found in Cyc and the way the knowledge base is used our work.

8.1.4 Controlled languages and sublanguages

In some natural language processing applications, restrictions on the lexicon and the knowledge base are actually deliberate. A reduced set of words, clearly defined grammar rules for a parser, and particular semantic interpretations for concepts in a knowledge base also lead to a reduction of lexical, syntactic, and semantic ambiguity.

Basic English (C. K. Ogden, 1933, 1934, 1935, 1937) was designed as an easy to learn second language and was originally intended facilitate the communication between native and non-native English speakers in science, business, and a variety of other fields. Its lexicon consisted of merely 850 words, 600 of them general and picturable ‘things’, 150 ‘qualities’, and 100 ‘operations’.[1] More words could be added for adopting the language to a particular domain. The guiding principle for the design of Basic English was the ‘elimination’ of the verb by reducing the number of verbs to a

[1] ‘Operations’ did not only include a minimal set of verbs but also pronouns, determiners, conjunctions and various other parts of speech.

small set of primitives.[2] The guiding principle for the design of the lexicon was 'one word, one meaning', i.e. each word in the lexicon had exactly one corresponding part of speech. This principle, also referred to as the 'golden rule', has been followed in the design of many other controlled languages that followed Basic English. The grammar itself is an extremely simplified version of Standard English, consisting of only five main rules. Although Basic English did not see any widespread use, it provided the foundation for the development of other controlled languages.

Export-oriented manufacturers of heavy machinery like Caterpillar, Boeing, or Scania picked up the idea of controlled languages to create service manuals for their products (Almquist & Sagvall Hein, 1996). The controlled language used in the document is intended to be effortlessly and unambiguously understood by the intended reader. Furthermore, the simplified source language can also be easily translated into different target languages. Several of these proprietary controlled languages led to the development of a manufacturer-independent language. The aerospace industry has developed AECMA Simplified English (AECMA, 1995) as a standard for the preparation of maintenance manuals intended to be used by native and non-native speakers of English. Similar to Basic English, Simplified English uses a reduced lexicon and a simplified grammar. Each entry in the lexicon exists only for one part of speech and has a single particular meaning. A number of validation tools exist for AECMA Simplified English including a controlled language syntax checker developed at Boeing (Wojcik, Hoard, & Holzhauser, 1990; Wojcik & Holmback, 1996).

The KANT system (Mitamura, Nyberg, & Carbonell, 1993; Nyberg & Mitamura, 1992) uses KANT Controlled English (Mitamura & Nyberg, 1995) for a knowledge-based translation of technical documents. It puts constraints on the source text by using a limited vocabulary with distinct word senses and a grammar that places restrictions on the syntactic complexity of the source. Although the initial design of the controlled language called for an unambiguous mapping between lexical entries and word senses, the development of Caterpillar Technical English and its use with the KANT system showed that the use of different word senses cannot be avoided in practical applications. KANT Controlled English and Caterpillar Technical English use a less 'fundamentalist' approach and allows multiple meanings for a word. Any lexical and semantic ambiguities are resolved by an interactive process during the creation of the document.

---

[2] Interestingly, the idea of using a small set of atomic actions surfaces again in early semantic theories for natural language processing such as Conceptual Dependency theory (Schank, 1975) and the LNR-style representations found in (Gentner, 1975).

Controlled languages are also used to provide a more natural way of specifying tasks and operating procedures for machinery such as ATMs. Attempto Controlled English (Fuchs, Schwertel, & Schwitter, 1999; Fuchs & Schwitter, 1996) is a controlled language for such specifications that can be directly transformed into logical forms. (Eijk, Koning, & Steen, 1996) provides a general overview of controlled languages, their use, and various implementation issues.

Information extraction systems often get their greatest leverage from highly domain-specific sublanguages (Kittredge & Lehrberger, 1982). These languages were not specifically designed for the information extraction task, but did already exist, in one form or another, as a formalized way of recording information like patient data, criminal reports, or other kinds of formalized information (Hirschman & Sager, 1982; Sager, 1982).

8.1.5 Parsing and Tagging

The parser described in chapter 5 makes use of the core parsing algorithm of the chart parser described in (Allen, 1995). This parser is a limited version of the one found in the TRAINS system (Allen et al., 1995; Traum et al., 1994). We enabled the parser to query the background knowledge base for general semantic information for terminal nodes. The semantic knowledge is combined in a bottom-up fashion at parse time when phrase nodes are constructed.

Integrated parsing approaches that combine syntactic and semantic information have been used in a variety of earlier systems such as IPP (Lebowitz, 1980), MOPTRANS (Lytinen, 1984), (Wilks, 1975b), ELI (Riesbeck & Schank, 1976), and DMAP (Riesbeck, 1986). A more recent system that uses integrated parsing for robust natural language understanding under ‘real-world’ conditions is ParseTalk (Hahn, Broeker, & Neuhaus, 2000). Its parser uses a depth-first approach and trades off completeness against efficiency. ParseTalk analyzes texts on a paragraph level instead of isolated sentences and provides the parser with conceptual domain-specific information as background knowledge in addition to lexical and syntactic information. The tight integration of semantic information from a knowledge base and the grammatical information is similar to the approach used in our system. The ParseTalk parser implements a rather sophisticated message passing approach with a number of protocol layers (Neuhaus & Hahn, 1996). If the basic protocol cannot completely analyze a sentence due to unknown words, the parser will try to ignore those unknown items by using a skipping protocol. Should this attempt also fail, the parser uses on a backtracking protocol and tries to attach at least partial information from the current sentence to the structures it has build from the preceding sentences. Using backtracking and skipping techniques the parser can deal with partial and sparse information as well as prosaic, overly specific information.

The TRAINS parser supports best-first parsing strategy, which ensures that parser returns the phrases that cover the longest sequence of words in the input text. If the parser cannot find a complete sentence structure, it will at least return any partial information it found for substructures such as completed noun and verb phrases. Partial parsing (Abney, 1991) is an important technique to recover as much information as possible from unrestricted text, without requiring a complete parse of the entire sentence. Partial parsers such as Fidditch (Hindle, 1994), Cass2 (Abney, 1996a) and Copsy (Schwarz, 1990) are not only robust in processing noisy input sentences but are also fast (Abney, 1996b). Because partial parsers analyze only structures that can be reliably identified, the output usually consists of a set of fragments instead of complete interpretations. These techniques are useful in information extraction tasks such as the Message Understanding Conference competitions (Grishman & Sundheim, 1996) where deep sentence understanding is traded off against a correct identification of particular phrase information used as template fillers.

The performance of the parser can often be improved by using a part-of-speech tagger prior to the actual parsing attempt. While most taggers make use of statistical information, one of the best and widely used part-of-speech taggers is the symbolic tagger described in (Brill, 1992). The tagger determines the most likely part of speech for each word in the input sentence and allows the parser to filter out potentially ambiguous lexicon entries that are not compatible with the part of speech information delivered by the tagger. Although extra time is spent on running the tagger and filtering, there are several benefits to this approach. The commitment to a single part of speech for a word eliminates lexical ambiguity in the input and avoids potential syntactic ambiguity arising from grammar rules firing on particular lexical choices. The time spent by the parser on creating incompatible chart entries and backtracking from incomplete phrase structures is usually greater than the time spent on the tagging and filtering operations. Moreover, the use of a tagger can improve the accuracy of the parser such that it generates better interpretations. (Charniak et al., 1996) provides an overview and comparison of different types of taggers and examines how the uses of a tagger affects the parsing results. (Macklovitch, 1992) analyzes a number of grammatical issues that are problematic for statistical taggers.

8.1.6 Semantic Interpretation

Semantic interpretation is a key component of nearly every natural language processing system. A semantic interpreter is not needed in systems that use a tightly controlled language. In addition to a grammar that eliminates potential syntactic ambiguity, these systems allow only one lexical entry for each word, and a single meaning per lexical entry. There is only one possible interpretation for each word in

every accepted syntactic variation of the input. While the semantic interpretation process in these systems is reduced to a trivial one-to-one mapping between words and their meaning, the limitations of the lexicon and the grammar make these systems difficult to use. The user has to know the sense for each word to avoid miscommunication, in addition to detailed knowledge about the exact grammar rules to ensure that the sentence is parsed correctly. Adding new entries to the lexicon is difficult, because any potential semantic overlap with existing entries has to be avoided. In practice, most natural language processing systems use lexicons that contain potentially ambiguous entries. Words can appear in different parts of speech and may even have multiple senses for the same speech part, each represented by individual lexicon entries.

The grammar for the restricted language described in chapter 5 is sensitive to syntactic structures that reflect patterns used for expressing information related to QP Theory in natural language. Tying syntactic structures to a particular semantic interpretation is a technique exemplified by Semantic Grammar (Burton, 1976a). It worked especially well in domains where tasks use a highly structured language, for example in tutoring systems such as SOPHIE (Brown & Burton, 1975) or for natural language database front-ends such as LIFER (Hendrix, 1977; Hendrix, Sacerdoti, Sagalowicz, & Slocum, 1978). Systems that solely use semantic grammars are highly dependent to their particular domain or even to a single task. LIFER uses different grammars for different types of databases, depending on their particular content. For this reason we have encoded only QP-specific patterns as grammar rules, but not any domain-dependent patterns.

The frame-based approach used in our system, i.e. the use of QP frame structures by the semantic interpreter to capture information extracted by the parser, borrows ideas from early frame-based NL systems such as GUS (Bobrow et al., 1977). Many other systems have employed such techniques in various forms, from early NLU systems such as POLITICS (Carbonell, 1979) or BORIS (Lehnert et al., 1983) to recent projects such as the KANT machine translation system and its interlingua representations (Nyberg & Mitamura, 1992). The conceptual analysis approach and its implementation CA (Birnbaum & Selfridge, 1981) use Conceptual Dependency structures (Schank, 1975) to capture information from natural language text. The top-down approach described in (Palmer, 1990) uses three different representation levels for the semantic interpretation process. The template level corresponds to syntactic realizations of sentential units, similar to the support of QP-specific patterns in our system. The canonical level just below the templates uses case frame representations and associates possible semantic roles with verb complements. The representations of the case frame level are then expanded to produce finer-grained expressions on the predicate level. Since this work took place during the early years of the Cyc project and predates the FrameNet, the knowledge base and the case frame representations

used in Palmer's system were hand-crafted and covered a small domain (pulley problems).

The semantic interpretation process in all of these systems would have greatly benefited from general, reusable resources such as FrameNet and the Cyc knowledge base. For example, Talmy's work, relating the structure of language to fundamental aspects of cognition such as space, time, and causality (Talmy, 2000), as well as the work on representations for event structures (Davidson & Harman, 1972; Parsons, 1990), situation semantics (Barwise & Perry, 1983) and the use of thematic roles (Dowty, 1991; Fillmore, 1976; Somers, 1987) are seminal ideas in semantics that have found their way into the design of modern knowledge bases such as Cyc. While the semantic interpretation process of our system makes use of these resources, QP Theory contributes inferential semantics for FrameNet information, as it allows standard qualitative reasoning techniques to be used on information from natural language text captured as frame data. Furthermore, we think that the qualitative mathematics of QP Theory can provide a formalism for Talmy's work on force and causation (Talmy, 1988).

Semantic interpretations that rely on commonsense world knowledge are generalizations of interpretations produced by domain-specific systems. A greater challenge is a system that allows metaphorical interpretations, such as the approaches described in (Barnden, Helmreich, Iverson, & Stein, 1994; Carbonell, 1982; Indurkhya, 1992; Martin, 1990). A system that uses commonsense knowledge can be enabled to interpret metaphors, perhaps by relying on a number of stored cases demonstrating how particular information about a situation had been used in the past. This technique would allow processing novel metaphors as well as already known uses. Understanding metaphors is a challenge to information extraction techniques, unless their specialized extraction templates allow a particular metaphorical interpretation.

8.2 Future work

Although the implemented system described in this thesis can produce interpretations of QP-related information from natural language text, there are a number of things that should be improved in future versions of the system. There are basically four major areas in which such improvements can be made: the background knowledge, the controlled language, including tools for the production and validation of the input documents, the syntactic parser, including the maintenance of the lexicon and the grammar, and the semantic interpreter. All of these areas are step on the way to create a system that can accumulate information about physical phenomena from natural language text and over time create models of the underlying physical processes.

8.2.1 The background knowledge

Our system uses a subset of the Cyc knowledge base for general background knowledge for each word in the input description. This data is a valuable resource for producing general semantic interpretations, but it lacks domain-specific content. The acquisition of additional domain-specific information is desirable to provide more depth to the semantic interpretation process.

We have also encountered some erroneous content in the knowledge base. The semantic interpretation process can be hampered by missing denotational information, switched argument positions, and other undesired effects. Although the semantic interpreter can filter out some of these inconsistencies, others are more difficult to detect and require corrections to the knowledge base contents. An interactive natural language based approach could also be a possible solution to fix these problems. Instead of working around inconsistent knowledge by covering up and filtering out potential problems, the interpreter could support a mode in which the user is prompted to resolve detected problems interactively. While the focus of our work is on the semantic interpretation process, better knowledge engineering tools will certainly make the task easier.

8.2.2 The controlled language

Like human languages, an artificial language is shaped by its users. The controlled language is therefore expected to change, to be modified, and to be extended. There are a number of interesting research questions associated with the development of the controlled language. For example, how tightly can the language be controlled and be still useable for general purposes, i.e. how far can the syntactic and semantic ambiguity be reduced without imposing severe limitations on its expressiveness? As it has been discussed in chapter 5, the tradeoff for controlled languages is expressiveness and flexibility versus ambiguity. A very restrictive controlled language will be difficult to use and probably not be accepted by its users, while a loosely controlled language contains too much ambiguity to produce good semantic interpretations. A refinement of QRG-CE has to consider these tradeoffs. User testing will play an important part to find out which restrictions are acceptable and which syntactic structures are desirable. It would be important to select a diverse user base for these experiments, i.e. users with different interest and varying degrees of expertise in their particular domain.

Another interesting research aspect would be the automatic detection of language patterns for general QP-related content and for domain-specific information. The corpus analysis in chapter 3 has been done by hand, which proved to be a rather tedious process. It should be possible to use pattern discovery algorithms similar to

ExDisco (Yangarber & Grishman, 2000a) to scan larger corpora for particular syntactic patterns. However, encoding these patterns as part of the QRG-CE grammar has to be done manually and with care, since the appropriate semantic characteristics have to be attached to the grammar rules for each pattern. Furthermore, new patterns could interfere with existing grammatical rules or might display other undesired side effects.

8.2.3 The parser

Several improvements could be made to the basic chart parser as well. The parser lacks a morphological component. Adding such a component would allow a lexicon with fewer entries, since morphological variants such as plurals of nouns, inflected verb forms and degrees for adjectives and adverbs can be produced from a basic lexical entry. The current lexicon contains all these additional forms as individual entries, doubling the number of entries of the original COMLEX source. Instead of simply matching each word in the input sequence against the lexicon entries, the morphological component would analyze the words, find the associated basic lexical entry, and augment it with the information from the morphological analysis. The tradeoff here is the processing time spent on the analysis and the generation of the appropriate lexical entry versus the extra space required for the expanded version of the lexicon that already contains all morphological variants.

A better integration of the QP-specific syntactic patterns into the parsing process would be another desirable feature. The current bottom-up parsing algorithm detects sentence-level patterns late in the parsing process and is not very efficient especially for longer sentences containing relative clauses and sentence-level substructures. Combining a top-down parsing step, similar to the techniques used in semantic grammar (Burton, 1976b), can mediate this problem. The top-down parser would act as a pre-processor that detects QP-specific patterns, splits a sentence into smaller structures for each constituent of the pattern, parses each part individually by using the existing bottom-up parsing algorithm, and then recombines the results at the pattern level. The advantages are an better separation of the specialized QP patterns from the grammar, a more reliable recognition of these patterns by using semantic constraints in the top-down processing step, and a more efficient use of the bottom-up parsing algorithm on smaller substructures.

Another improvement would be a development of lexicon building tools. Such tools could scan a corpus for unknown words, e.g. technical terms of a particular domain, and then build new lexical entries based on the context in which these words were found. This strategy has been used in combination with HPSG-style grammars (Erbach, 1990; Kilbury et al., 1992). To generate the corresponding conceptual information in the background knowledge base, an interesting approach would be the

combination of lexicon building techniques with model of creative reading such as (Moorman & Ram, 1994).

8.2.4 The semantic interpreter

One problem not handled by the semantic interpretation process in our system is the generation of quantified structures. (Woods, 1978) and (Cooper, 1983) describe compositional approaches to quantifier scoping. Cooper uses a matrix for combing quantified phrases with the rest of the sentence and a quantifier store to keep track of the scooping potential of the quantifiers in the phrase. A more general and formalized version of this approach is the incremental interpretation framework in the Candide system for knowledge acquisition (Pereira & Pollack, 1991).

The paragraph-level interpretation process in its present form uses a simple merge algorithm to find overlapping information about quantities. This merge technique is similar to the mechanism used in a study on deriving semantic networks from controlled text descriptions with the KANT system (Nyberg et al., 2002). The next version of the semantic interpreter should include anaphora resolution and the detection of co-referential structures to improve the readability of the document. Repeating entity names across sentences to ensure that the semantic interpreter can correctly identify the quantities associated with these entities would no longer be necessary (Kamp, 1981; Kennedy & Boguraev, 1996; Zadrozny & Jensen, 1991).

Furthermore, the semantic interpretation process is focused on descriptions of process instances. The information extracted from a description and the models built by the semantic interpreter are stored as concrete, individual examples, e.g. the description of a heat flow process describes a particular instance. The semantic interpreter does not distinguish between general and exemplar-specific knowledge. However, generic process models might be abstracted from a collection of instances by using tools such as SEQL (Kuehne, Forbus, Gentner, & Quinn, 2000) for similarity-based generalization and abstraction. SEQL takes a number of exemplars, e.g. the representations of process descriptions, as its input and produces generalized descriptions of these exemplars. Information that is specific to an instance such as discourse variables and names of entities is stripped away and replaced by generic information. Sufficiently similar descriptions will be grouped together in one or more categories. SEQL even works on a diverse set of exemplars, i.e. descriptions of different types of processes mixed with other pieces of information, and performs robustly if subjected to noisy input (Kuehne, Gentner, & Forbus, 2000).

8.3 Outlook

These are interesting and exciting times for working in deep semantic natural language processing! It is time for a renaissance of a line of research that had been widely abandoned in favor of information extraction and retrieval techniques. A number of valuable computational, ontological, and linguistic resources have become available during the last decade, due to the efforts of large-scale projects such as Cyc, WordNet, and FrameNet. We are now at a point where these resources can be combined to attempt natural language understanding at a level that was not possible during the first wave of deep semantic processing that started thirty years ago.

References

Abney, S. (1991). Parsing by Chunks. In R. Berwick, S. Abney & C. Tenny (Eds.), *Principle-based Parsing*. Dordrecht, Netherlands: Kluwer Academic Publishing.

Abney, S. (1996a). *Partial Parsing via finite-state cascades.* In Proceedings: European Summer School in Logic, Language and Information, Workshop on Robust Parsing (ESSLLI 96), Prague, Czech Republic.

Abney, S. (1996b). Part-of-Speech Tagging and Partial Parsing. In K. Church, S. Young & G. Bloothooft (Eds.), *Corpus-based Methods in Language and Speech*. Dordrecht, Netherlands: Kluwer Academic Publishing.

AECMA. (1995). *A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language* (Simplified English document No. PSC-85-16598). Brussels, Belgium: Association Europeene des Constructeurs de Materiel Aerospatial.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence, 23*(2), 123-154.

Allen, J. F. (1995). *Natural Language Understanding* (2nd ed.). Redwood City, CA: Benjamin/Cummings.

Allen, J. F. (1998). *The TRAINS Parsing System, Version 4.0, A User's Manual*.Unpublished manuscript, Rochester, NY.

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P. A., Hwang, C.-H., Kato, T., et al. (1995). The TRAINS Project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI (JETAI), 7*, 7-48.

Almquist, I., & Sagvall Hein, A. (1996). *Defining Scania Swedish - a Controlled Language for Truck Maintenance.* In Proceedings: First International Workshop on Controlled Language Applications (CLAW-96), University of Leuven, Belgium.

Androutsopoulos, I., Ritchie, G. D., & Thanisch, P. (1995). Natural Language Interfaces to Databases - An Introduction. *Journal of Language Engineering, 1*(1), 29-81.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). *The Berkeley FrameNet Project.* In Proceedings: 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 98), Montreal, Canada.

Bareiss, R. (1989). *Exemplar-Based Knowledge Acquisition, A Unified Approach to Concept Representation, Classification, and Learning*. San Diego, CA: Academic Press.

Barg, P., & Walther, M. (1998). *Processing Unknown Words in HPSG.* In Proceedings: Thirty-Sixth Annual Meeting of the Association for Computational Linguistics.

Barker, K. (1994). *Clause-level Relationship Analysis in the TANKA System* (Tech Report No. TR-94-07). Ottawa, Canada: Department of Computer Science, University of Ottawa.

Barker, K. (1996). *The assessment of semantic cases using English positional, prepositional and adverbial case markers* (Tech Report No. TR-96-01). Ottawa, Canada: Department of Computer Science, University of Ottawa.

Barker, K. (1997). *Noun modifier relationship analysis in the TANKA system* (Tech Report No. TR-97-02). Ottawa, Canada: Department of Computer Science, University of Ottawa.

Barker, K. (1998). *Semi-Automatic Recognition of Semantic Relationships in English Technical Texts.* Ph.D. thesis, University of Ottawa, Ottawa, Canada.

Barker, K., Delisle, S., & Szpakowicz, S. (1998). *Test-driving TANKA: Evaluating a Semi-Automatic System of Text Analysis for Knowledge Acquisition.* In Proceedings: Canadian Conference on Artificial Intelligence.

Barker, K., & Szpakowicz, S. (1995). *Interactive semantic analysis of Clause-Level Relationships.* In Proceedings: Second Conference of the Pacific Association for Computational Linguistics., Brisbane, Australia.

Barnden, J. A., Helmreich, L., Iverson, E., & Stein, G. C. (1994). *An integrated implementation of simulative, uncertain, and metaphorical reasoning about mental states.* In Proceedings: Fourth International Conference on Principles of KR and Reasoning, Bonn, Germany.

Barwise, J., & Perry, J. (1983). *Situations and attitudes*. Cambridge, Mass.: MIT Press.

Bateman, J. A. (1993). *Ontology construction and natural language.* In Proceedings: Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation, Padova.

Bierwisch, M. (1967). Some semantic universals of German adjectivals. *Foundations of Language, 3*, 1-36.

Bierwisch, M. (1989). The Semantics of Gradation. In M. Bierwisch & E. Lang (Eds.), *Dimensional Adjectives* (pp. 71-261). Berlin, Germany: Springer-Verlag.

Bikel, D. M., Miller, S., & Weischedel, R. M. (1997). *Nymble: a high-performance learning name-finder.* In Proceedings: Fifth Conference on Applied Natural Language Processing (ANLP-97), Washington, DC.

Bikel, D. M., Schwartz, R. L., & Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine Learning, 34*(1-3), 211-231.

Birnbaum, L., & Selfridge, M. (1981). Conceptual Analysis of Natural Language. In R. C. Schank & C. K. Riesbeck (Eds.), *Inside Computer Understanding* (pp. 318-353). Hillsdale, NJ: Lawrence Erlbaum Associates.

Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., & Winograd, T. (1977). GUS, A Frame-Driven Dialog System. *Artificial Intelligence, 8*(2), 155-173.

Brill, E. (1992). *A simple rule-based part-of-speech tagger.* In Proceedings: Third Conference on Applied Natural Language Processing (ANLP-92), Trento, Italy.

Brill, E. (1994). *Some Advances in Transformation-Based Part of Speech Tagging.* In Proceedings: Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, WA.

Brown, J. S., & Burton, R. R. (1975). Multiple Representations of Knowledge for Tutorial Reasoning. In D. G. Bobrow & A. Collins (Eds.), *Representation and Understanding* (pp. 311-349). Orlando, FL: Academic Press, Inc.

Buckley, S. (1979). *From Sun Up to Sun Down*. New York: McGraw-Hill.

Burns, K. J., & Davis, A. R. (1999). Building and Maintaining a semantically adequate Lexicon using Cyc. In E. Viegas (Ed.), *Breadth and Depth of Semantic Lexicons* (pp. 121-143). Dordrecht, The Netherlands: Kluwer Academic Publishing.

Burton, R. R. (1976a). *Semantic Grammar: A technique for efficient language understanding in limited domains.* Doctoral dissertation, University of California, Irvine, CA.

Burton, R. R. (1976b). *Semantic Grammar: An engineering technique for constructing natural language understanding systems* (No. BBN Report No. 3453). Cambridge, MA: Bolt Beranek and Newman, Inc.

Califf, M. E. (1998). *Relational Learning Techniques for Natural Language Information Extraction* (Tech Report No. AI98-276). Austin, TX: University of Texas.

Califf, M. E., & Mooney, R. J. (1998). *Relational Learning of Pattern-Match Rules for Information Extraction.* In Proceedings: AAAI Spring Symposium on Applying Machine Learning to Discourse Processing, Stanford, CA.

Carbonell, J. G. (1979). *Subjective understanding: Computer models of belief systems* (Research report No. 150). New Haven, CT: Yale University.

Carbonell, J. G. (1982). Metaphor Comprehension. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for Natural Language Processing* (pp. 413-433). Hillsdale, NJ.

Cardie, C. (1994). *Domain-specific knowledge acquisition for conceptual sentence analysis.* doctoral dissertation, University of Massachusetts, Amherst, MA.

Charniak, E. (1972). *Toward a model of children's story comprehension* (Technical Report No. AITR-266). Cambridge, MA: MIT.

Charniak, E., Carroll, G., Adcock, J., Cassandra, A., Gotoh, Y., Katz, J., et al. (1996). Taggers for Parsers. *Artificial Intelligence, 85*(1-2), 45-57.

Clark, P., & Matwin, S. (1992). *Learning Domain Theories using Abstract Background Knowledge* (No. TR-92-95). Ottawa: University of Ottawa, Canada.

Clark, P., & Matwin, S. (1993). *Using Qualitative Models to Guide Inductive Learning.* In Proceedings: 10th International Machine Learning Conference (ML-93).

Clark, P., & Porter, B. (1997). *Building Concept Representations from Reuseable Components.* In Proceedings: AAAI 97.

Clark, P., Thompson, J., & Porter, B. (2000). *Knowledge Patterns.* In Proceedings: KR 2000.

Cooper, R. (1983). *Quantification and syntactic theory*. Dordrecht, Holland: D. Reidel Pub. Co.

Copeck, T., Barker, K., Delisle, S., Szpakowicz, S., & Delannoy, J.-F. (1997). What is Technical Text? *Language Science*, *19*(4), 391-424.

Copestake, A., & Sparck Jones, K. (1990). Natural Language Interfaces to Databases. *Knowledge Engineering Review, 5*(4), 225-249.

Cullingford, R. E. (1978). *Script application: Computer understanding of newspaper stories* (Research report No. 116). New Haven, CT: Yale University.

Dahlgren, K. (1988). *Naive Semantics for Natural Language Understanding*. Boston, MA: Kluwer Academic Publishers.

Davidson, D., & Harman, G. (Eds.). (1972). *Semantics of Natural Language*. Dordrecht, The Netherlands: D. Reidel Publishing Co.

Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P., & Vilain, M. (1997). *Mixed-Initiative Development of Language Processing Systems.* In Proceedings: Fifth Conference on Applied Natural Language Processing, Association for Computational Linguistics, Washington, D.C.

DeJong, G. F. (1979). *Skimming stories in real time: An experiment in integrated understanding* (Research report No. 158). New Haven, CT: Yale University.

DeJong, G. F. (1982). An Overview of the FRUMP System. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for natural language processing* (pp. 149-176). Hillsdale, NJ: Erlbaum.

Delisle, S., & Szpakowicz, S. (1995). *Realistic Parsing: Practical Solutions of Difficult Problems.* In Proceedings: Second Conference of the Pacific Association for Computational Linguistics., Brisbane, Australia.

Dixon, R. M. W. (1991). *A New Approach to English Grammar, on Semantic Principles*. Oxford, England: Clarendon Press.

Dowty, D. R. (1991). Thematic proto-roles and argument selection. *Language, 67*(3), 547-619.

Eijk, P. v. d., Koning, M. d., & Steen, G. v. d. (1996). *Controlled language correction and translation.* In Proceedings: First International Workshop on Controlled Language Applications (CLAW-96), University of Leuven, Belgium.

Erbach, G. (1990). Syntactic Processing of Unknown Words. In P. Jorrand & V. Sgurev (Eds.), *Artificial Intelligence IV - Methodology, Systems, Applications*. Amsterdam, The Netherlands: North-Holland.

Everett, J. O. (1999). Topological inference of teleology: Deriving function from structure via evidential reasoning. *Artificial Intelligence, 113*(1-2), 149-202.

Falkenhainer, B., Farquhar, A., Bobrow, D., Fikes, R., Forbus, K., Gruber, T., et al. (1994). *CML: A Compositional Modeling Language*.(Technical Report KSL-94-16). Stanford, CA: Stanford University, Knowledge Systems Laboratory.

Fellbaum, C. (Ed.). (1998). *WordNet - an electronic lexical database*. Cambridge, MA: MIT Press.

Fillmore, C. J. (1968). The Case for Case. In E. Bach & R. T. Harris (Eds.), *Universals in linguistic theory* (pp. 1-88). New York: Holt, Rinehart and Winston.

Fillmore, C. J. (1976). Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, 280*, 20-32.

Fillmore, C. J., & Atkins, B. T. S. (1994). Starting where the dictionaries stop: The challenge for computational lexicography. In B. T. S. Atkins & A. Zampolli (Eds.), *Computational Approaches to the Lexicon*. New York: Oxford University Press.

Fillmore, C. J., & Baker, C. F. (2001). *Frame Semantics for Text Understanding.* In Proceedings: WordNet and other lexical resources workshop, NAACL, Pittsburgh, PA.

Fillmore, C. J., Wooters, C., & Baker, C. F. (2001). *Building a Large Lexical Databank Which Provides Deep Semantics.* In Proceedings: Pacific Asian Conference on Language, Information, and Computation, Hong Kong, China.

Forbus, K. D. (1984). Qualitative Process Theory. *Artificial Intelligence, 24*, 85-168.

Forbus, K. D., & Falkenhainer, B. (1990). *Self-explanatory simulations: An integration of qualitative and quantitative knowledge.* In Proceedings: 8th National Conference on Artifical Intelligence (AAAI-90), Boston, MA.

Forbus, K. D., & Kleer, J. d. (1993). *Building Problem Solvers*. Cambridge, MA: MIT Press.

Forbus, K. D., Mostek, T. A., & Ferguson, R. W. (2002). *An analogy ontology for integrating analogical processing and first-principles reasoning.* In Proceedings: Fourteenth Innovative Applications of Artificial Intelligence Conference (IAAI '02), Edmonton, Alberta, Canada.

Frawley, W. (1992). *Linguistic Semantics*. Hillsdale, NJ: Erlbaum.

Fuchs, N. E., Schwertel, U., & Schwitter, R. (1999). Attempto Controlled English - Not just another logic specification language. In P. Flener (Ed.), *Logic-based Program Synthesis and Transformation, 8th International Workshop LOPSTR-98*. Manchester, England: Springer Verlag.

Fuchs, N. E., Schwertel, U., & Torge, S. (1999). *Controlled Natural Language can replace First-Order Logic.* In Proceedings: 14 IEEE International Conference on Automated Software Engineering (ASE-99), Cocoa Beach, FL.

Fuchs, N. E., & Schwitter, R. (1996). *Attempto Controlled English (ACE).* In Proceedings: First International Workshop on Controlled Language Applications (CLAW-96), University of Leuven, Belgium.

Gentner, D. (1975). Evidence for the Psychological reality of Semantic Components: The Verbs of Possession. In D. A. Norman & D. E. Rumelhart (Eds.), *Explorations in Cognition* (pp. 211-246). San Francisco: Freeman.

Grishman, R. (1997). *Information Extraction: Techniques and Challenges.* In Proceedings: International Summer School, SCIE-97, Frascati, Italy.

Grishman, R., & Sundheim, B. (1996). *Message Understanding Conference - 6: A Brief History*. In Proceedings: 16th International Conference on Computational Linguistics, Copenhagen, Denmark.

Gritzen, E. F. (Ed.). (1980). *Introduction to Naval Engineering.* Annapolis, MD: Naval Institute Press.

Guha, R. V., & Lenat, D. B. (1990). Cyc: A Midterm Report. *AI Magazine, 11*(3), 32-59.

Hahn, U., Broeker, N., & Neuhaus, P. (2000). Let's ParseTalk: Message-passing protocols for object-oriented parsing. In H. Bunt & A. Nijholt (Eds.), *Advances in Probabilistic and other Parsing Technologies* (pp. 177-201). Dordrecht, Netherlands: Kluwer Academic Publ.

Hahn, U., & Schnattinger, K. (1998). *A text understander that learns.* In Proceedings: 17th International Conference on Computational Linguistics (COLING '98) and 36th Annual Meeting of the Association for Computational Linguistics (ACL '98), Montreal, Quebec, Canada.

Hammitt, G. M. (1997). *Learn Spanish the fast and fun way* (2nd ed.). New York, NY: Barron's Educational Series, Inc.

Hayes, P. J. (1985). Naive Physics I: Ontology for Liquids. In J. R. Hobbs & R. C. Moore (Eds.), *Formal Theories of the Commonsense World* (pp. 71-89). Norwood, NJ: Ablex.

Hendrix, G. G. (1977). *LIFER: A Natural Language Interface Facility.* In Proceedings: Second Berkeley Workshop on Distributed Data Management and Computer Networks, University of California, Berkeley, CA.

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., & Slocum, J. (1978). Developing a Natural Language Interface to Complex Data. *Transactions on Database Systems, 3*(2), 105-147.

Hindle, D. (1994). A parser for text corpora. In B. T. S. Atkins & A. Zampolli (Eds.), *Computational Approaches to the Lexicon.* Oxford, England: Oxford University Press.

Hirschman, L., Light, M., Breck, E., & Burger, J. D. (1999). *Deep Read: A Reading Comprehension System.* In Proceedings: 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, MD.

Hirschman, L., & Sager, N. (1982). Automatic Information Formatting of a Medical Sublanguage. In R. Kittredge & J. Lehrberger (Eds.), *Sublanguage* (pp. 27-80). Berlin, Germany: Walter de Gruyter & Co.

Hirst, G. (1987). *Semantic interpretation and the resolution of ambiguity*. Cambridge, England: Cambridge University Press.

Hobbs, J. R., Appelt, D., Bear, J., Israel, D., Kameyama, M., Strickel, M., et al. (1996). FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In *Finite State Devices for Natural Language Processing*. Cambridge, MA: MIT Press.

Huddleston, R. D. (1971). *The sentence in written English: A syntactic study based on an analysis of scientific texts*. Cambridge, England: Cambridge University Press.

Huddleston, R. D. (1984). *Introduction to the grammar of English*. Cambridge, England: Cambridge University Press.

Indurkhya, B. (1992). *Metaphor and Cognition, an interactionist approach*. Boston, MA: Kluwer.

Johnson, C. R., Fillmore, C. J., Wood, E. J., Ruppenhofer, J., Urban, M., Petruck, M. R. L., et al. (2001). *The FrameNet Project: Tools for Lexicon Building*.Unpublished manuscript, Berkeley, CA.

Kamp, H. (1981). A theory of truth and semantic interpretation. In J. A. G. Groenendijk, T. M. V. Janssen & M. J. B. Stokhof (Eds.), *Formal methods in the study of language* (pp. 277-322). Amsterdam, Netherlands: Mathematisch Centrum.

Katz, J. J., & Fodor, J. A. (1963). The structure of semantic theory. *Language, 39*, 170-210.

Kennedy, C. (2000). *Scalar Representations in Natural Language Semantics*.Unpublished manuscript.

Kennedy, C. (2001). Polar Opposition and the Ontology of 'Degrees'. *Linguistics and Philosophy, 24*(1), 33-70.

Kennedy, C., & Boguraev, B. (1996). *Anaphora for everyone: Pronominal anaphora resolution without a parser.* In Proceedings: 16th International Conference on Computational Linguistics (COLING '96), Copenhagen, Denmark.

Kennedy, C., & McNally, L. (1999). *Degree modification and the scalar structure of gradable adjectives.* In Proceedings: Description des Adjectifs pour les Traitements Informatiques, Cargese, France.

Kilbury, J., Naerger, P., & Renz, I. (1992). *New Lexical Entries for Unknown Words*.Unpublished manuscript, Universitaet Duesseldorf, Germany.

Kittredge, R., & Lehrberger, J. (Eds.). (1982). *Sublanguage, Studies of language in restricted semantic domains*. Berlin, Germany: Walter de Gruyter.

Knight, K. (1996). *Learning Word Meanings by Instruction.* In Proceedings: National Conference on Artificial Intelligence (AAAI '96).

Kuehne, S. E. (2003). *On the Representation of Physical Quantities in Natural Language.* In Proceedings: Seventeenth International Workshop on Qualitative Reasoning (QR '03), Brasilia, Brazil.

Kuehne, S. E., & Forbus, K. D. (2002). *Qualitative Physics as a component in natural language semantics: A progress report.* In Proceedings: Twentyfourth Annual Conference of the Cognitive Science Society, George Mason University, Fairfax, VA.

Kuehne, S. E., Forbus, K. D., Gentner, D., & Quinn, B. (2000). *SEQL: Category Learning as Progressive Abstraction using Structure Mapping.* In Proceedings: Twentysecond Annual Conference of the Cognitive Science Society, Institute for Research in Cognitive Science, Philadelphia, PA.

Kuehne, S. E., Gentner, D., & Forbus, K. D. (2000). *Modeling Infant Learning via Symbolic Structural Alignment.* In Proceedings: Twentysecond Annual Conference of the Cognitive Science Society, Institute for Research in Cognitive Science, Philadelphia, PA.

Lebowitz, M. (1980). *Generalization and memory in an integrated understanding system* (Research report No. 186). New Haven, CT: Yale University.

Lehnert, W., Dyer, M. G., Johnson, P. N., Yang, C. J., & Harley, S. (1983). BORIS - An Experiment in In-Depth Understanding of Narratives. *Artificial Intelligence, 20*(1), 15-62.

Lehr, P. E., Burnett, R. W., & Zim, H. S. (1987). *Weather*. New York, NY: Golden Books Publishing Company, Inc.

Lenat, D. B., & Guha, R. V. (1989). *Building large knowledge-based systems : representation and inference in the Cyc project*. Reading, MA: Addison-Wesley.

Lytinen, S. L. (1984). *The Organization of Knowledge in a Multilingual Integrated Parser*. Ph.D. thesis, Yale University, New Haven, CT.

Macklovitch, E. (1992). *Where the Tagger Falters.* In Proceedings: Fourth Conference on Theoretical and Methodological Issues in Machine Translation.

Macleod, C., Grishman, R., & Meyers, A. (1998). *COMLEX Syntax Reference Manual, Version 3.0*. Philadelphia, PA: Linguistic Data Consortium, University of Pennsylvania.

Mahesh, K., & Nirenburg, S. (1995). *A Situated Ontology for Practical NLP*. In Proceedings: IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal.

Mars, N. J. I. (1993). *An ontology of measurement units.* In Proceedings: International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation, Padova, Italy.

Martin, J. H. (1990). *A Computational Model of Metaphor Interpretation*. Cambridge, MA: Academic Press.

Maton, A., Hopkins, J., Johnson, S., LaHart, D., McLaughlin, C. W., Warner, M. Q., et al. (1994). *Heat Energy* (annotated teacher's ed.). Englewood Cliffs, NJ: Prentice Hall.

Matwin, S., & Rouget, T. (1996). *Explainable Induction with an Imperfect Qualitative Model*.Unpublished manuscript, Ottawa, Canada.

Mellish, C. S. (1985). *Computer Interpretation of Natural Language Descriptions*. New York, NY: Halsted Press.

Minsky, M. (1975). A Framework for Representing Knowledge. In P. H. Winston (Ed.), *The Psychology of Computer Vision*. New York, NY: McGraw-Hill.

Mitamura, T., & Nyberg, E. H. (1995). *Controlled English for Knowledge-Based MT: Experience with the KANT System.* In Proceedings: 6th International Conference on Theoretical and Methodological Issues in Machine Translation, Leuven, Belgium.

Mitamura, T., & Nyberg, E. H. (2001). *Automatic Rewriting for Controlled Language Translation.* In Proceedings: 6th Natural Language Pacific Rim Symposium, Workshop on Automatic Paraphrasing: Theories and Applications, Tokyo, Japan.

Mitamura, T., Nyberg, E. H., & Carbonell, J. G. (1993). *Automated Corpus Analysis and the Acquisition of Large, Multi-Lingual Knowledge Bases for MT.* In Proceedings: Fifth International Conference on Theoretical and Methodological Issues in Machine Translation, Kyoto, Japan.

Mooney, R. J. (1987). *Integrated learning of words and their underlying concepts.* In Proceedings: 9th Annual Conference of the Cognitive Science Society (CogSci '87), Seattle, WA.

Moorman, K., & Ram, A. (1994). *A Functional Theory of Creative Reading* (Technical Report GIT-CC-94/01). Atlanta, GA: Georgia Institute of Technology.

Moran, J. M., & Morgan, M. D. (1994). *Meteorology - The Atmosphere and the Science of Weather* (4th ed.). New York, NY: Macmillan College Publishing.

Neuhaus, P., & Hahn, U. (1996). *Trading off Completeness for Efficiency - The PARSETALK Performance Grammar Approach to Real-World Parsing.* In Proceedings: 9th Florida Artificial Intelligence Research Symposium (FLAIRS '96).

Niles, I., & Pease, A. (2001a). *Origins of the IEEE Standard Upper Ontology.* In Proceedings: 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Workshop on the IEEE Standard Upper Ontology, Seattle, WA.

Niles, I., & Pease, A. (2001b). *Towards a Standard Upper Ontology.* In Proceedings: Second International Conference on Formal Ontology (FOIS 2001), Ogunquit, MA.

Noy, N. F., & Hafner, C. D. (1997). The State of the Art in Ontology Design. *AI Magazine, 18,* 53-74.

Nyberg, E. H., Kamprath, C., & Mitamura, T. (1998). The KANT Translation System: From R&D to Large-Scale Deployment. *Localization Industry Standards Association Newsletter, 2*(1).

Nyberg, E. H., & Mitamura, T. (1992). *The KANT System: Fast, accurate, high-quality translation in practical domains.* In Proceedings: 15th International Conference on Computational Linguistics (COLING 92), Nantes, France.

Nyberg, E. H., & Mitamura, T. (1996). *Controlled Language and Knowledge-Based Machine Translation: Principles and Practice.* In Proceedings: First International Workshop on Controlled Language Applications, Leuven, Belgium.

Nyberg, E. H., Mitamura, T., Baker, K., Svoboda, D., Peterson, B., & Williams, J. (2002). *Deriving Semantic Knowledge from Descriptive Texts using an MT System.* In Proceedings: 5th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2002), Tiburon, CA.

Nyberg, E. H., Mitamura, T., & Carbonell, J. (1997). *The KANT Machine Translation System: From R&D to Initial Deployment.* In Proceedings: Localization Industry Standards Association Workshop on Integrating Advanced Translation Technology, Washington, D.C.

Ogden, C. K. (1933). *Basic by Examples*. London: K. Paul, Trench, Trubner and Co., Ltd.

Ogden, C. K. (1934). *The Basic Dictionary* (3rd ed.). London: K. Paul, Trench, Trubner, and Co., Ltd.

Ogden, C. K. (1935). *Basic English versus the artificial languages*. London: K. Paul, Trench, Trubner and Co., Ltd.

Ogden, C. K. (1937). *The system of Basic English*. New York: Harcourt, Brace and Company.

Ogden, W. C., & Bernick, P. (1997). Using Natural Language Interfaces. In M. G. Helander, T. K. Landauer & P. V. Prabhu (Eds.), *Handbook of Human-Computer Interaction* (pp. 137-161). Amsterdam: Elsevier Science Publishers B.V.

Palmer, M. S. (1990). *Semantic processing for finite domains*. Cambridge, England: Cambridge University Press.

Panton, K., Miraglia, P., Salay, N., Kahlert, R. C., Baxter, D., & Reagan, R. (2002). *Knowledge Formation and Dialog Using the KRAKEN Toolset.* In Proceedings: Fourteenth Conference on Innovative Applications of Artificial Intelligence (IAAI '02), Edmonton, Alberta, Canada.

Parsons, T. (1990). *Events in the Semantics of English*. Cambridge, MA: MIT Press.

Pazienza, M. T. (Ed.). (1997). *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, SCIE-97, Frascati, Italy* (Vol. 1299). Berlin: Springer Verlag.

Pazienza, M. T. (Ed.). (1999). *Information Extraction: Towards Scalable, Adaptable Systems* (Vol. 1714). Berlin: Springer Verlag.

Pease, A., Niles, I., & Li, J. (2002). *The Suggested Upper Merged Ontology: A large ontology for the Semantic Web and its applications.* In Proceedings: 18th National Conference on Artificial Intelligence (AAAI 02), Workshop on Ontologies and the Semantic Web, Edmonton, Canada.

Pereira, F. C. N., & Pollack, M. E. (1991). Incremental interpretation. *Artificial Intelligence, 50*(1), 37-82.

Petruck, M. R. L. (1996). Frame Semantics. In J. Verschueren, J.-O. Oestman, J. Blommaert & C. Bulcaen (Eds.), *Handbook of Pragmatics*. Philadelphia, PA: John Benjamins.

Pollard, C. J., & Sag, I. A. (1994). *Head-driven phrase structure grammar*. Stanford: Center for the Study of Language and Information, University of Chicago Press.

Quirk, R. (1985). *A Comprehensive grammar of the English language*. London, England: Longman.

Raskin, V., & Nirenburg, S. (1995). *Lexical Semantics of Adjectives: A Microtheory of Adjectival Meaning* (Technical Report No. MCCS-95-288). Las Cruces, NM: New Mexico State University.

Riesbeck, C. K. (1986). From Conceptual Analyzer to Direct Memory Access Parsing: An Overview. In N. E. Sharkey (Ed.), *Advances in Cognitive Science* (Vol. 1, pp. 236-258). New York, NY: Halsted Press.

Riesbeck, C. K., & Schank, R. C. (1976). *Comprehension by Computer: Expectation-based Analysis of Sentences in Context* (Technical Report No. 78). New Haven, CT: Yale University, Department of Computer Science.

Riloff, E. (1993). *Automatically Constructing a Dictionary for Information Extraction Tasks.* In Proceedings: 11th National Conference on Artificial Intelligence (AAAI-93), Washington, DC.

Riloff, E. (1996). *Automatically Generating Extraction Patterns from Untagged Text.* In Proceedings: 13th National Conference on Artificial Intelligence (AAAI-96), Portland, OR.

Ritchie, G. D., Russell, G. J., Black, A. W., & Pulman, S. G. (1991). *Computational Morphology*: MIT Press.

Sager, N. (1982). Syntactic Formatting of Science Information. In R. Kittredge & J. Lehrberger (Eds.), *Sublanguage* (pp. 9-26). Berlin, Germany: Walter de Gruyter & Co.

Schank, R. C. (1975). Conceptual dependency theory. In R. C. Schank (Ed.), *Conceptual Information Processing* (pp. 22-82). Amsterdam, Netherlands: North-Holland.

Schank, R. C., & Tesler, L. (1969, September 1-4, 1969). *A Conceptual Dependency Parser for Natural Language.* In Proceedings: International Conference on Computational Linguistics (COLING-69), Sanga-Saeby, Sweden.

Schmidt, F. W., Henderson, R. E., & Wolgemuth, C. H. (1993). *Introduction to Thermal Sciences* (2nd ed.). New York, NY: John Wiley & Sons, Inc.

Schnattinger, K., & Hahn, U. (1997). *Intelligent text analysis for dynamically maintaining and updating domain knowledge bases.* In Proceedings: 2nd International Symposium on Advances in Intelligent Data Analysis: Reasoning about Data, London, England.

Schnattinger, K., & Hahn, U. (1998). *Quality-based learning.* In Proceedings: 13th European Conference on Artificial Intelligence (ECAI 98), Brighton, England.

Schwarz, C. (1990). Automatic syntactic analysis of free text. *Journal of the American Society for Information Science (JASIS), 41*(6), 406-417.

Schwitter, R., & Fuchs, N. E. (1996). *Attempto - From Specifications in Controlled Natural Language towards Executable Specifications.* In Proceedings: EMISA Workshop 'Natuerlichsprachlicher Entwurf von Informationssystemen - Grundlagen, Methoden, Werkzeuge, Anwendungen', Ev. Akademie, Tutzing, Germany.

Scott, S., & Matwin, S. (1998). *Using Lexical Knowledge in Text Classification* (Tech. Report No. TR-98-03). Ottawa, Canada: University of Ottawa.

Simpson, J. A., & Weiner, E. S. C. (Eds.). (1989). *Oxford English Dictionary* (2nd ed.). Oxford, England: Oxford University Press.

Soderland, S. (1999). Learning Information Extraction Rules for Semi-structured and Free Text. *Machine Learning, 34*(1-3), 233-272.

Soderland, S., Fisher, D., Aseltine, J., & Lehnert, W. (1995). *CRYSTAL: Inducing a Conceptual Dictionary.* In Proceedings: Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada.

Somers, H. L. (1987). *Valency and Case in Computational Linguistics.* Edinburgh, Scotland: Edinburgh University Press.

Staab, S., Erdmann, M., & Maedche, A. (2000). *Semantic Patterns* (No. Technical Report 412). Karlsruhe, Germany: AIFB, Universitaet Karlsruhe.

Talmy, L. (1988). Force dynamics in language and cognition. *Cognitive Science, 12*(1), 49-100.

Talmy, L. (2000). *Toward a cognitive semantics*. Cambridge, MA: MIT Press.

Traum, D. R., Allen, J. F., Ferguson, G., Heeman, P. A., Hwang, C.-H., Kato, T., et al. (1994, 21-23 March 1994). *Integrating Natural Language Understanding and Plan Reasoning in the TRAINS-93 Conversation System.* In Proceedings: AAAI Spring Symposium on Active NLP, Stanford, CA.

Verduijn, A. N. (2002). *The new language in international business - Simplified English*. Tilburg, The Netherlands: Tedopres International B.V.

Vilain, M., & Day, D. (1996). *Finite-State Parsing by Rule Sequences.* In Proceedings: International Conference on Computational Linguistics (COLING 96), Copenhagen, Denmark.

Voorhees, E. M. (1999). *Natural Language Processing and Information Retrieval.* In Proceedings: International Summer School, SCIE-99, Rome, Italy.

Watt, W. C. (1968). Habitability. *American Documentation, 19*(3), 338-351.

Wiemer-Hastings, P., Graesser, A. C., & Wiemer-Hastings, K. (1998). *Inferring the Meaning of Verbs from Context.* In Proceedings: 20th Annual Conference of the Cognitive Science Society, Madison, WI.

Wilks, Y. (1975a). An intelligent analyzer and understander of English. *Communications of the ACM, 18*(5), 264-274.

Wilks, Y. (1975b). A Preferential Pattern-matching Semantics for Natural Language Understanding. *Artificial Intelligence, 6*(1), 53-74.

Wilks, Y. (1997). *Information Extraction as a Core Language Technology.* In Proceedings: International Summer School, SCIE-97, Frascati, Italy.

Wilks, Y., & Stevenson, M. (1996). *The grammar of sense: Is word sense tagging much more than part-of-speech tagging?* (Technical Report No. CS-96-05). Sheffield, England: University of Sheffield.

Williams, J. (1992). *The Weather Book*. New York, NY: USA Today, Random House.

Winograd, T. (1972). *Understanding natural language*. Edinburgh,: Edinburgh University Press.

Wisniewski, E. J., & Gentner, D. (1991). On the combinatorial semantics of noun pairs: minor and major adjustments to meaning. In G. B. Simpson (Ed.), *Understanding word and sentence* (pp. 241-284). Amsterdam, Netherlands: North-Holland.

Wisniewski, E. J., & Murphy, G. L. (1989). Superordinate and basic category names in discourse: A textual analysis. *Discourse Processes, 12*(2), 245-261.

Wojcik, R. H., Hoard, J. E., & Holzhauser, K. C. (1990). *The Boeing Simplified English Checker*. In Proceedings: International Conference on Human Machine Interaction and Artificial Intelligence in Aeronautics and Space, Centre d'Eude et de Recherche de Toulouse, France.

Wojcik, R. H., & Holmback, H. (1996). *Getting a Controlled Language off the ground at Boeing.* In Proceedings: First International Workshop on Controlled Language Applications (CLAW 96), Leuven, Belgium.

Wojcik, R. H., Holmback, H., & Hoard, J. (1998). *Boeing Technical English: An Extension of AECMA SE beyond the Aircaft Maintenance Domain.* In Proceedings: Second International Workshop on Controlled Language Applications (CLAW 98), Pittsburgh, PA.

Wolff, P. (2003). Direct causation in the linguistic coding and individuation of causal events. *Cognition, 88*(1), 1-48.

Wolff, P., Song, G., & Driscoll, D. (2002). *Models of causation and causal verbs.* In Proceedings: 37th Meeting of the Chicago Linguistics Society, Chicago, IL.

Woods, W. A. (1978). Semantics and Quantification in Natural Language Question Answering. In B. J. Grosz, K. Sparck Jones & B. L. Webber (Eds.), *Readings in Natural Language Processing* (pp. 205-248). Los Altos, CA: Morgan Kaufman Publishers.

Woods, W. A., Kaplan, R. M., & Nash-Webber, B. (1972). *The Lunar Sciences Natural Language Information System, Final Report* (BBN Report No. 2378). Cambridge, MA: Bolt Beranek and Newman, Inc.

Yangarber, R., & Grishman, R. (1997). *Customization of Information Extraction Systems.* In Proceedings: International Workshop on Lexically Driven Information Extraction, Frascati, Italy.

Yangarber, R., & Grishman, R. (2000a). *Extraction Pattern Discovery through Corpus Analysis.* In Proceedings: Workshop ' Information Extraction meets Corpus Linguistics', Second International Conference on Language Resources and Evaluation (LREC 2000), Athens, Greece.

Yangarber, R., & Grishman, R. (2000b). *Machine Learning of Extraction Patterns from Unannotated Corpora: Position Statement.* In Proceedings: Workshop 'Machine Learning for Information Extraction', 14th European Conference on Artificial Intelligence (ECAI 2000), Berlin, Germany.

Zadrozny, W., & Jensen, K. (1991). Semantics of Paragraphs. *Computational Linguistics*, *17*(2), 171-209.

Appendix A

Natural language patterns for QP constituents

A.1 Patterns for Indirect Influences

Based on roughly 100 sentences from the corpus material (Buckley, 1979; Maton et al., 1994; Moran & Morgan, 1994) we classified information about indirect influences and isolated a set of distinct patterns.

A.1.1 II1: THE x-er/THE y-er

Pattern:

```
THE <Comparative1> <Quantity1> [<Change1>],
 THE <Comparative2> <Quantity2> [<Change2>].
```

Instantiations:

(1) "The bigger the thermal resistance, the harder it is for heat to flow, since the resistance to the flow of heat is increased."

Comparative1: bigger
Quantity1: thermal resistance
Change1: -
Comparative2: harder
Quantity2: heat
Change2: flows

(2) "The larger the surface area is, the more convection heat is lost from the surface."

Comparative1: larger
Quantity1: (surface) area
Change1: -
Comparative2: more
Quantity2: heat
Change2: lost

(3) "The bigger the surface is, the more heat flows from the surface."

Comparative1: bigger
Quantity1: (surface) area
Change1: -

Comparative2: more
Quantity2: heat (of surface)
Change2: flows (from)

(4) "The greater the heating, the greater the bending."
Comparative1: greater
Quantity1: heating
Change1: -
Comparative2: greater
Quantity2: bending
Change2: -

(5) "In addition, the greater the temperature difference (that is, the steeper the temperature gradient), the more rapid is the rate of heat."
Comparative1: greater
Quantity1: temperature difference
Change1: -
Comparative2: more rapid
Quantity2: heat (flow rate)
Change2: -

(6) "The faster an object moves, the more kinetic energy it has."
Comparative1: faster
Quantity1: (speed) object
Change1: moves
Comparative2: more
Quantity2: kinetic energy (object)
Change2: -

(7) "The faster you swing the hammer, the farther the nail is driven into the wood."
Comparative1: faster
Quantity1: (speed) hammer
Change1: swing
Comparative2: farther
Quantity2: (depth) nail
Change2: driven

(8) "The higher the temperature of a substance, the faster the molecules in that substance are moving, on the average."
Comparative1: higher
Quantity1: temperature of substance
Change1: -

Comparative2: faster
Quantity2: (speed) molecules
Change2: move

(9) "The colder the winter, the more potholes in the spring!"
Comparative1: colder
Quantity1: (temp of) winter
Change1: -
Comparative2: more
Quantity2: (amount of) potholes
Change2: -

Characterization:
<Quantity> is a phrase that contains information about the quantity type and the entity it is associated with. For most of the examples listed here, the <Quantity> is actually just the quantity type. For example, ‘temperature’ by itself is not a quantity. The pattern would also allow the use of phrase such as ‘the temperature of the brick’, which contains sufficient information to construct a quantity frame.

For (1) we can argue that 'heat' is not actually the quantity type heat, but refers to the superordinate heat flow process. The quantity is the flowrate as determined by the verb 'flow' and 'harder', with a negative sign. Alternatively we can treat 'heat' as a quantity type indeed and interpret 'flows harder' as a negative sign. In both cases, we have to interpret the action and the comparative.

In (2) and (3) we face a similar problem. Is 'heat' a quantity to which we apply a sign directly, or is it an object which has a particular quantity type? Sentences (2) and (3) can refer to an increase of the heat flow rate, or simply a decrease in heat (as indicated by 'more lost' and 'more flows from').

In (4) 'heating' and 'bending' can be thought of as abbreviation for the quantities 'heating rate' and 'bending rate'. Sentence (9) is a somewhat odd example. 'Winter' really means the quantity 'winter temperature' and 'potholes' is actually is the number (or amount) of potholes.

Sentences like (5), (6), (7) or (8) are easy to interpret. In sentence (6), the combination of the verb 'move' and the adverb 'faster' refers to the quantity type 'velocity' of an entity.

A.1.2 II2: AS x, y

Patterns:

```
AS <Quantity1> <Change1>, <Quantity2> <Change2>.
<Quantity1> <Change1>, AS <Quantity2> <Change2>.
```

Instantiations:

(1) "As the air temperature rises and heat is transferred to the thermometer, the liquid expands and rises in tile glass tube."

Quantity1: temp. of air
Change1: rises
Quantity2: (volume) liquid
Change2: expands

(2) Quantity1: heat of thermometer
Change1: transferred to
Quantity2: (volume) liquid
Change2: expands

(3) "As the air and thermometer cool, the liquid contracts and drops in the tube."

Quantity1: (temp) air, thermometer
Change1: cools
Quantity2: (volume) liquid
Change2: contracts

(4) "As heat is supplied to the bulb, the fluid (usually mercury) expands upward and beyond the constriction."

Quantity1: heat of bulb
Change1: supplied
Quantity2: (volume) fluid
Change2: expands

(5) "As the temperature rises again, the fluid expands and the index is left behind at the lowest (minimum) temperature."

Quantity1: temp (fluid)
Change1: rises
Quantity2: (volume) fluid
Change2: expands

(6) "However, as wind speed increases, the thickness of the boundary layer diminishes, and the rate of sensible heat loss from the body increases."

Quantity1: speed of wind
Change1: increases
Quantity2: thickness of layer
Change2: diminishes

(7)
Quantity1: speed of wind
Change1: increases
Quantity2: rate of sensible heat
Change2: increases

(8) "As the molecules move faster, they move farther apart."

Quantity1: (speed) molecules
Change1: move faster
Quantity2: (distance of molecules)
Change2: move farther apart

(9) "Remember that as a liquid is heated, its molecules move faster and farther apart."

Quantity1: (temp) liquid
Change1: heated
Quantity2: (distance, speed) molecules
Change2: move faster, farther apart

(10) "So as the liquid in a thermometer gets warmer, it expands and rises in the tube."

Quantity1: (temp) liquid
Change1: gets warmer
Quantity2: (volume) liquid
Change2: expands

(11) "As heat is added to the liquid water after the phase change, the temperature rises again until it reaches 100°C."

Quantity1: heat of water
Change1: added
Quantity2: temperature (of water)
Change2: rises

(12) "As heat energy is added to the solid, the kinetic energy of the molecules increases and their vibrations speed up."

Quantity1: heat energy of solid
Change1: added
Quantity2: kinetic energy of molecules

Change2: increases

Quantity1: heat energy of solid
Change1: added
Quantity2: (speed of molecules)
Change: speed up

(13) "As the volume increases, the density decreases."
Quantity1: volume (of substance)
Change1: increases
Quantity2: density (of substance)
Change2: decreases

(14) "As the temperature of a gas increases, the molecules move faster and faster."
Quantity1: temp. of gas
Change1: increases
Quantity2: (speed) molecules
Change2: move faster

Characterization:
This pattern also appears in a 'reversed' form, like "The temperature rises as heat is added." The reversed form is less often used than the standard form. Again, the associated entity can be omitted from the <Quantity>.

In (1) the first part describes the quantity 'temperature' of the entity 'air' with a positive change determined by the <Change> 'rises'. The second part identifies the entity 'liquid' but leaves out the quantity type. The verb 'expand' here refers indirectly to a positive change in a quantity type associated with liquid, volume.

In (2) the first part of the sentence mentions a quantity type ('heat') but leaves out the entity that the quantity type belongs to. The <Change> 'transferred to thermometer' describes a positive change of the quantity. However, the quantity type 'heat' does not really belong to the thermometer but to the liquid contained in it. This entity is then mentioned in the second part of the sentence.

The first part of (3) is similar to the latter part of (1). Both 'air' and 'liquid' are entities, the verbs 'cool' and 'contract' refer indirectly to a change in quantities associated with air and liquid. The quantity they refer to and the sign of derivative are determined by the verb. In both cases, the sign is negative, but 'cools' refers to a change in temperature, while 'contracts' refers to a change in volume.

Sentence (4) is similar to (2). In the first part, we have an isolated quantity type ('heat') but no entity it is associated with. The <Change> 'supplied' refers to a positive change in the quantity.

Sentence (13) is apparently a clean and easy case: both 'volume' and 'density' are quantity types, with direct (positive - 'increases', and negative - 'decreases') changes. Nevertheless, there is a subtle difference between the quantities involved here. Volume is an extensive quantity and can be directly influenced (i.e. volume, or mass) can be added); however, density is not an extensive quantity (i.e. we cannot take away density from an object).

The first part of (15) is a variation of (1). The quantity type ('temperature') and the entity it belongs to ('gas') are both mentioned, and the verb ('increases') indicates a positive change. In the second part, only the object is mentioned, and the quantity (velocity) needs to be determined by the verb 'move' and the adverb 'fast'.

As in the examples of the contracting liquid, the cooling air and the expanding fluid, the verb (and the adverb) will be the lemma of a frame. The verb 'move' belongs to the motion domain, and the adverb 'fast' (or 'slow') will highlight a particular aspect of movement, velocity.

A.1.3 II3: WHEN x, y

Patterns:

```
<Quantity1> <Change1>, WHEN <Quantity2> <Change2>
WHEN <Quantity1> <Change1>, <Quantity2> <Change2>
```

Instantiations:

(1) "When a liquid or gas is heated, the molecules begin to move faster."

Quantity1:	(speed) molecules
Change1:	move faster
Quantity2:	(temp) liquid/gas
Change2:	heated

(2) "The liquid in a thermometer expands when it is heated."

Quantity1:	(volume) liquid
Change1:	expands
Quantity2:	(temp) liquid
Change2:	heated

(3) "Most substances - solids, liquids, and gases - expand when their temperature is increased."

Quantity1: (volume) substances
Change1: expand
Quantity2: temperature (of substances)
Change2: increase

(4) "The kinetic energy of the molecules in a liquid also increases when the liquid is heated."

Quantity1: kinetic energy of molecules
Change1: increased
Quantity2: (temp) liquid
Change2: is heated

(5) "This equation shows why the density of water changes when its volume changes."

Quantity1: density of water
Change1: changes
Quantity2: volume (of water)
Change2: changes

Characterization:

The standard WHEN pattern is a variation of the reversed AS pattern. They can be used interchangeably. Like the AS pattern, there is also a less often used 'reversed' form of the WHEN pattern, i.e. WHEN <Quantity> <Change>, <Quantity> <Change>, and mirrors the standard AS pattern.

A.1.4 II4: VERB PATTERNS

A.1.4.1 DEPENDS ON

Pattern:

```
<Quantity1> DEPENDS ON  <Quantity2>
```

Instantiations:

(1) "We've already learned that it depends in the first place on temperature difference (as does all heat flow) as well as on the kind of material involved (whether conductor or insulator)."

Quantity1: (heat flow rate)
Quantity2: temperature difference

(2) "It also depends on the area of the heat-flow path."
Quantity1: (heat flow rate)
Quantity2: area of path

(3) "Heat flows from surfaces at a rate depending on the size of the surface."
Quantity1: heat flow rate
Quantity2: size of surface

(4) "How fast it flows depends on how wide you open the valve."
Quantity1: how fast it flows (flow rate)
Quantity2: how wide valve opened (area of flow path)

(5) "The temperature response of a substance that gains or loses heat depends on the specific heat of that substance."
Quantity1: temperature response of substance
Quantity2: specific heat of substance

(6) "The amount of heat produced depends on the amount of motion."
Quantity1: amount of heat
Quantity2: amount of motion

(7) "Unlike temperature, heat depends on the mass of the substance present."
Quantity1: heat of substance
Quantity2: mass of substance

Characterization:
This pattern is more complex than the previous three, since it is less constrained and allows a greater variation of what precedes and follows the 'depends on' keyword. There are also no clear indicators signaling a change in a quantity.

With the exception of sentence 4, the pattern uses no verbs other than ‘depends on’. Because of the fact that the pattern does not include the sign for a change, it can only express a qualitative proportionality in its weakest form, i.e. that there is an influence of one quantity on another, without actually stating whether the influence is positive or negative.

This a variation of the DEPENDS ON pattern in the form of the CHANGES WHEN/WITH patterns. These two have a similar form:

```
<Quantity1> CHANGES WITH <Quantity2>
<Quantity1> CHANGES WHEN <Quantity2> CHANGES.
```

The entities/things do not include the sign of the change; they simply say that one quantity changes with another one in an unspecified direction.

A.1.4.2 AFFECTS

Patterns:

```
<Quantity1> [Sign] AFFECTS <Quantity2>
<Quantity1> AFFECTS <Quantity2> [Sign]
```

Instantiations:

(1) "In addition, radiation heat flow can be affected by how shiny the surface is, whereas convection cannot."

Quantity1:	radiation heat flow
Quantity2:	how shiny the surface is
Sign:	-

(2) "But depth difference affects the volume flow: if the tank isn't full, less will flow out."

Quantity1:	depth difference
Quantity2:	volume flow
Sign:	-

(3) "Third, flow-path area affects volume flow; there is less flow through a thin pipe than a fat one."

Quantity1:	flow-path area
Quantity2:	volume flow
Sign:	-

Characterization:

The first quantity in (1) is actually the flow rate of the radiation heat flow process, the second quantity is the smoothness of the surface of an unspecified object.

Sentence (2) is a little more complicated. The first quantity, the depth difference is actually composed of two individual quantities, i.e. (- (depth obj1) (depth obj2)). The second quantity here is again the flowrate of the volume flow process. An alternative interpretation could be that this is not a qualitative proportionality but a correspondence like (Q= flowrate (- (depth obj1) (depth obj2))).

A.1.4.3 INFLUENCES

Patterns:

```
<Quantity1> [Sign] INFLUENCES <Quantity2>
<Quantity1> INFLUENCES <Quantity2> [Sign]
```

Instantiations:

(1) "Second, the speed at which the convection gas or liquid flows by the surface influences how quickly the heat flows."

Quantity1:	speed of convection flow
Quantity2:	speed of heat flow
Sign:	-

A.1.4.4 CAUSES

An extension of the AFFECTS and INFLUENCES patterns is the CAUSES pattern in which both quantities appear together with a change indicator.

Pattern:

```
<Change1> <Quantity1> CAUSES <Change2> <Quantity2>
```

Instantiations:

(1) "In circumstances where heat gain and heat loss affect the temperature, net heat gain causes air temperature to rise, whereas net heat loss causes air temperature to drop."

Change1:	gain
Quantity1:	heat of air
Change2:	rise
Quantity2:	temperature of air

Change1:	loss
Quantity1:	heat of air
Change2:	drop
Quantity2:	temperature of air

(2) "When fast-moving molecules collide with slow-moving molecules, heat energy is transferred from the faster molecules to the slower molecules, causing the slower molecules to move faster."

Change1:	transferred to
Quantity1:	heat energy (molecules)
Change2:	move faster

Quantity2: (velocity of molecules)

(3) "Thermal expansion is the expansion, or increase in size, of a substance caused by heat."
Change1: (increase)
Quantity1: heat (of substance)
Change2: increase
Quantity2: size (of substance)

Characterization:
Sentence (1) includes two instances of the pattern, for the effects of both an increase and decrease in heat. The quantities and the changes are clearly identifiable, even though they are realized in a compound noun. Other possible 'subpatterns' are 'a decrease in heat' or 'decreasing heat'. Also, the participant to which the quantity type belongs might be mentioned or not. In sentence (1), both quantity types ('heat' and 'temperature') belong to the same participant, 'air' (which is only mentioned together with 'temperature').

In (2), the changes to the quantities are not explicitly expressed as a gain, increase, or loss. Similar to the other patterns above we have to derive the change from the verb (in this case, 'transferred to' would mean an increase. We also have to derive the second quantity (velocity) from the verb (move faster).

Sentence (3) is actually a 'reversed' subpattern. An unspecified change in a quantity causes the increase in size. However, an alternative interpretation of this sentence might be a causal relationship between two processes, i.e. heating causes expansion.

The change indicators (such as 'a gain in ...', 'a decrease in …') are not optional. For example, we cannot say 'Heat causes temperature.' A quantity cannot be the cause for another. However, a change in a quantity can be that cause for a change in another, as in 'A gain in heat causes an increase in temperature'. The indicators are/tend to be nouns that are directly tied to a particular sign, such as 'gain', 'loss', 'increase', 'decrease'. For these reasons, sentence (3) is a bad example (and might actually lead us to consider the alternative interpretation.)

There is a sign-neutral form of this pattern, however. For example, 'A change in heat causes a change in temperature'. Even if just one of indicator is neutral, the overall expression is neutral, as demonstrated in 'A gain in heat causes a change in temperature'. We simply cannot determine whether the gain in heat will have a positive or negative effect on the temperature. Additionally, we also have the problem that we do not know the objects associated with the quantity types.

A.2 Patterns for Direct Influences

Direct influences describe causal effects on a quantity. The combination of these effects will then determine the dynamic changes on a quantity. From our corpus material, we have extracted more than 60 sentences that we have marked as those containing information about direct influences. The following is a discussion of the most frequent syntactic patterns used for realizing statements about direct influences.

Statements about Direct Influences include two quantities, the influenced quantity (i.e. the constrained quantity) and the influencing (or, constraining) quantity. For example, in `(I+ (heat obj) (flowrate heatflow))`, the heat of an object is positively influenced by the flowrate of a heat flow process, i.e. the object obj gains heat. Furthermore, it must be noted that in the analyzed corpus material only one sentence mentions an influencing quantity, i.e. a rate of transfer between two quantities.

A.2.1 DI1: Transfer between quantities (active voice)

Pattern:

```
<QType> <Change> [from <Entity1>] [to <Entity2>]
                 [via <Path>]
```

Instantiations:

(1) "Both volume and heat flow - both can move from place to place."

QType:	volume
Change:	moves
Entity1:	place
Entity2:	place
Path:	-

QType:	heat
Change:	moves
Entity 1:	place
Entity 2:	place
Path:	-

(2) "Similarly, heat flows downhill from a higher temperature to a lower one."

QType:	heat
Change:	flows
Entity 1:	higher temperature
Entity 2:	lower one
Path:	-

(3) "Heat flows from hot things to cold things."

QType:	heat
Change:	flows
Entity 1:	hot things
Entity 2:	cold things
Path:	-

(4) "When you pour water out of a pitcher into a glass, volume flows from the pitcher to the glass."

QType:	volume
Change:	flows
Entity 1:	pitcher
Entity 2:	glass
Path:	-

(5) "When you open a door on a cold day, heat flows from inside the house to the outdoors."

QType:	heat
Change:	flows
Entity 1:	inside the house
Entity 2:	outdoors
Path:	door

(6) "If two cans having different depths are connected by a tube, volume will always flow toward the depth that is lower."

QType:	volume
Change:	will flow
Entity 1:	-
Entity 2:	lower depth
Path:	-

(7) "Even so, volume flows toward the can with the lower depth."

QType:	volume
Change:	flows
Entity 1:	-
Entity 2:	lower depth
Path:	-

(8) "If a small, hot stone is put into a big pan of warm water, heat will flow out of the stone and into the water, since the stone is hotter than the water."

QType:	heat
Change:	will flow

Entity 1: hotter stone
Entity 2: water
Path: -

(9) "Since the chicken started out colder than the refrigerator, heat had to flow into it to get it to be as 'warm' as the refrigerator."
QType: heat
Change: had to flow
Entity 1: refrigerator
Entity 2: chicken
Path: -

(10) "If we then put the chicken into a hot oven, more heat would flow into it until it became as hot as the oven."
QType: heat
Change: would flow
Entity 1: oven
Entity 2: chicken
Path: -

(11) "By analogy, the same volume flows from a hole in a can whether the can is on the floor or a table."
QType: volume
Change: flows
Entity 1: can
Entity 2: -
Path: hole

(12) "For instance, a silver spoon is hot when you've been stirring coffee because heat flows easily through the silver from the hot coffee to your fingers."
QType: heat
Change: flows
Entity 1: hot coffee
Entity 2: fingers
Path: silver (spoon)

(13) "How quickly the ice melts will measure how much heat is flowing through the bar from the coffee."
QType: heat
Change: is flowing

Entity 1: coffee
Entity 2: -
Path: bar

(14) "Infrared film, for example, is sensitive to radiation heat flow; it is used to photograph heat leaving objects by radiation."
QType: heat
Change: leaving
Entity 1: objects
Entity 2: -
Path: (radiation)

(15) "That is, heat flows from locations of higher temperature toward locations of lower temperature."
QType: heat
Change: flows
Entity 1: locations of higher temperature
Entity 2: locations of lower temperature
Path: -

(16) "Radiation is also the principal means whereby heat escapes from the planet to space."
QType: heat
Change: escapes
Entity 1: planet
Entity 2: space
Path: (radiation)

(17) "The ice cube in your hand is melting because heat is moving from your hand to the ice cube."
QType: heat
Change: is moving
Entity 1: hand
Entity 2: ice cube
Path: -

(18) "If you have ever accidentally touched a hot pan, you have discovered for yourself (most likely in a painful way) that heat energy moves from a warmer object to a cooler object."
QType: heat energy
Change: moves
Entity 1: warmer object

Entity 2: cooler object
Path: -

(19) "Because the water is warmer than the ice, heat moves from the water to the ice."
QType: heat
Change: moves
Entity 1: water
Entity 2: ice
Path: -

Characterization:
The <QType> is the type of quantity that is transferred, while <Change> is an active motion verb such as 'flows', 'moves'. The last three parts of the pattern are optional. In many cases both of the entities are mentioned as from- and to-locations, in rare instances also the path. From-location is indicated by a variety of prepositions such as 'from', 'out of', 'away from'. The same is true for to-location which uses the prepositions 'to', 'toward', 'into' etc.

A.2.2 DI2: Transfer between quantities (passive voice)

Pattern:

```
<QType> <Change> [by <Agent>] [from <Entity1>]
                 [to <Entity2>] [via <Path>]
```

Instantiations:
(1) "But instead of a surface heating the liquid or gas, previously heated liquid or gas is transported, or moved, from one place to another."
QType: (liquid/gas)
Change: is transported
Agent: -
Entity1: one place
Entity 2: another
Path: -

(2) "As heat is supplied to the bulb, the fluid (usually mercury) expands upward and beyond the constriction."
QType: heat
Change: is supplied
Agent: -
Entity 1: -
Entity 2: bulb
Path: -

(3) "As the more energetic molecules of the hot coffee collide with the less energetic atoms of the cooler spoon, some kinetic energy is transferred to the atoms of the spoon."

QType:	kinetic energy
Change:	is transferred
Agent:	-
Entity 1:	-
Entity 2:	atoms of the spoon
Path:	-

(4) "Heat is conducted from warm ground to cooler overlying air, but because air has a low heat conductivity, conduction is significant only in a very thin layer of air that is in immediate contact with the Earth's surface."

QType:	heat
Change:	is conducted
Agent:	-
Entity 1:	warm ground
Entity 2:	cooler overlying air
Path:	-

(5) "As heat is conducted from the relatively warm ground to cooler overlying air, the air becomes warmer than the surrounding air."

QType:	heat
Change:	is conducted
Agent:	-
Entity 1:	bottom of file pan
Entity 2:	water
Path:	-

(6) "When fast-moving molecules collide with slow-moving molecules, heat energy is transferred from the faster molecules to the slower molecules, causing the slower molecules to move faster."

QType:	heat energy
Change:	is transferred
Agent:	-
Entity 1:	faster molecules
Entity 2:	slower molecules
Path:	-

(7) "You know that when you cook soup or boil water, heat energy must be added to the liquid in order to raise its temperature."

QType:	heat energy
Change:	must be added
Agent:	-
Entity 1:	-
Entity 2:	liquid
Path:	-

Characterization:
In almost all instances, the <QType> is the type of the influenced quantity. One exception is sentence (1), in which the moved entity is a substance. The associated quantity type might be volume in this case. <Change> is a passive motion verb describing an action done by an optional. The last three parts of the pattern are optional. In many cases both entities are mentioned as from- and to-locations, in rare instances also a path. From-location is indicated by a variety of prepositions such as 'from', 'out of', 'away from'. This is the 'reverse' version of Pattern 1 (except for the optional <Agent>)

A.2.3 DI3: Explicitly mentioned transfer event

Patterns:

```
<QType> <Change> [from <Entity1>] [to <Entity2>]
<Change> <QType> [from <Entity1>] [to <Entity2>]
```

Instantiations:

(1) "The direction of volume flow is always downhill--from a higher depth to a lower one."

QType:	volume
Change:	flow
Entity1:	higher depth
Entity 2:	lower depth

(2) "Heat flow is always toward the object with the lower temperature - in this case, the water."

QType:	heat
Change:	flow
Entity 1:	-
Entity 2:	lower temperature

Characterization:
The <QType> is the type of influenced quantity, while <Change> is a noun, e.g. flow, associated with the <Quantity>.

A.2.4 DI4: Quantity change in object (active voice)

Pattern:

```
<Agent> <Change> <QType> [from <Entity1>]
                         [to <Entity2>]  [<Path>]
```

Instantiations:

(1) "Similarly, a full can of water will leak volume from a hole in the side of the can."

Agent:	can
Change:	leaks
QType:	volume
Entity 1:	-
Entity 2:	-
Path:	hole

(2) "For example, you feel cold on a windy day because the wind carries heat away from your skin by convection heat transfer."

Agent:	wind
Change:	carries
QType:	heat
Entity 1:	skin
Entity 2:	-
Path:	-

(3) "You feel warm in front of a fireplace mostly because the flames and hot coals move heat to your skin by radiation."

Agent:	flames
Change:	move
QType:	heat
Entity 1:	-
Entity 2:	skin
Path:	radiation

(4) "To understand this, think again of the example of a house losing heat through its chimney."

Agent:	house
Change:	loses
QType:	heat

Entity 1: -
Entity 2: -
Path: chimney

(5) "Rather, radiation is the principal means whereby the Earth-atmosphere system gains heat from the sun."
Agent: Earth-atmosphere system
Change: gains
QType: heat
Entity 1: sun
Entity 2: -
Path: (radiation)

(6) "Air in motion increases the rate of sensible heat loss (the combined effect of conduction and convection) from the body."
Agent: air in motion
Change: increases
QType: heat loss
Entity 1: body
Entity 2: -
Path: (conduction), (convection)

(7) "Sitting near an open fire, you know that the fire gives off heat."
Agent: fire
Change: gives off
QType: heat
Entity 1: -
Entity 2: -
Path: -

(8) "Other familiar forms of heat transfer by radiation include the heat you can feel around an open fire or a candle flame, the heat near a hot stove, and the heat given off by an electric heater."
Agent: fire/flame/stove/heater
Change: gives off
QType: heat
Entity 1: -
Entity 2: -
Path: -

Characterization:
The <Agent> is the entity that possesses a particular quantity type. The <Change> is an active verb that indicates a change in the quantity, and the <Path> can be a physical path (e.g. through the chimney) or a process (e.g. by radiation) indicating an implicit path.

A.2.5 DI5: Quantity change in object (passive voice)

Patterns:

```
<QType> <Change>    by <Agent>
<QType> <PosChange> by/to <Agent> [from <Entity>]
<QType> <NegChange> by/from <Agent> [to <Entity>]
```

Instantiations:

(1) "The heat gained or lost by a substance is equal to the product of its mass times the change in temperature (deltaT) times its specific heat."

QType:	heat
Change:	gained, lost
Agent:	substance
Entity:	-

(2) "Within a closed container, the heat lost by one substance must equal the heat gained by another substance."

QType:	heat
Change:	gained, lost
Agent:	substance
Entity:	-

(3) "Because the heat given off by the chemical reaction equals the heat gained by the water, the heat of the chemical reaction can be calculated."

QType:	heat
Change:	gained
Agent:	water
Entity:	-

QType:	heat
Change:	given off
Agent:	chemical reaction
Entity:	-

(4) "When ice melts and changes into water, energy in the form of heat is being absorbed by the ice."

QType: heat energy
Change: is being absorbed
Agent: water
Entity: -

Characterization:
The <Agent> is the entity that possesses a particular quantity type. The <Change> is a passive verb that indicates a change in the quantity. This is the 'reverse' version of Pattern 4 with two possible subpatterns that include to/from-style information about the second location/entity affected by the change. These two subpatterns cannot be combined, i.e. the from and to information is mutually exclusive.

A.3 Patterns for Ordinal Relations

A.3.1 OR1, OR2: Difference comparison between quantities

This pattern just states that a quantity associated with two objects is not equal. It does not give any ordering information, i.e. it does not state which object has the larger quantity. We have identified 4 subpatterns for generic differences between quantities.

A.3.1.1 OR1: Difference between quantities, noun subpattern 1

Pattern:

```
<QType> DIFF/N  [between <Entity1> and <Entity2>]
```

Instantiations:

(1) "The temperature difference - the brick's temperature minus the room's temperature - drives the heat from the brick."

QType: temperature
Entity1: (brick)
Entity2: (room)

(2) "The depth of the water is higher than the depth of the hole, so the depth difference drives volume out through the hole."

QType: depth
Entity1: -
Entity2: -

(3) "Only a depth difference (not volume) can cause volume to flow, just as only a temperature difference (not heat) can cause heat to flow."
QType: temperature
Entity1: -
Entity2: -

(4) "In this example, heat flow depends only on the temperature difference between the chicken and its surroundings, not on whether the surroundings are hot or cold."
QType: temperature
Entity1: chicken
Entity2: surroundings

(5) "Only the depth difference between the water level and the hole is important, not whether the can itself is raised or lowered."
QType: depth
Entity1: water level
Entity2: hole

(6) "We've already learned that it depends in the first place on temperature difference (as does all heat flow) as well as on the kind of material involved (whether conductor or insulator)."
QType: temperature
Entity1: -
Entity2: -

(7) "But depth difference affects the volume flow: if the tank isn't full, less will flow out."
QType: depth
Entity1: -
Entity2: -

(8) "One factor, we've already learned, is temperature difference."
QType: temperature
Entity1: -
Entity2: -

(9) "In convection heat flow, it's the temperature difference between the surface and the convecting gas or liquid that's important."
QType: temperature
Entity1: -
Entity2: -

(10) "As with all heat flow, radiation heat flow depends on temperature difference."
QType: temperature
Entity1: -
Entity2: -

(11) "The important temperature difference is that which the surface sees."
QType: temperature
Entity1: -
Entity2: -

(12) "As in conduction, convection, and radiation, the amount of heat flow depends on the temperature difference."
QType: temperature
Entity1: -
Entity2: -

(13) "In this case, the important temperature difference is that between the incoming airflow and the outgoing airflow - the temperature of the hot air going out the chimney minus that of the cold air leaking in through the cracks in the house."
QType: temperature
Entity1: incoming airflow
Entity2: outgoing airflow

Characterization:
The <QType> is the quantity type, e.g. temperature, depth etc. associated with the entities. DIFF/N is a noun expressing the inequality. In all instances of this pattern found in our corpus, the noun 'difference' was used. The two entities are optional. If they are not specified, contextual information must be used.

A.3.1.2 OR2: Difference between quantities, noun subpattern 2

Pattern:

```
DIFF/N  in <QType> [between <Entity1> and <Entity2>]
```

Instantiations:
(1) "Just as a difference in temperature causes heat to flow, so a difference in depth causes volume to flow."
QType: temperature
Entity1: -
Entity2: -

QType: depth
Entity1: -
Entity2: -

(2) "Convection occurs within the atmosphere as a consequence of differences in air density."
QType: (air) density
Entity1: -
Entity2: -

Characterization:
This pattern is a variation of noun subpattern 1. The <QType> is the quantity type, e.g. temperature, depth etc. associated with the entities. DIFF/N is a noun expressing the difference. In all instances of this pattern found in our corpus the noun 'difference' was used. The two entities are optional. If they are not specified, contextual information must be used. Theoretically, the sum of two quantities could also appear as a pattern, possibly indicated by the noun ‘sum’ or ‘combination’. However, the analyzed corpus material did not include any instance for this pattern.

A.3.1.3 OR2: Difference between quantities, adj. subpattern 1

Pattern:

```
<QType> <Entity1> [and <Entity2>] DIFF/ADJ
```

Instantiations:
(1) "Heat flows from one place to another because the temperature of the two places is different."
QType: temperature
Entity1: one place
Entity2: another

Characterization:
The <QType> is a quantity type, e.g. temperature, depth etc. associated with the entities participating in the ordinal relation. DIFF/ADJ is an adjective expressing the inequality. For all instances of this pattern occurring in our corpus material the word 'different' was used. One of the two entities is optional. We could have variations like 'the temperature is different' (no entities), 'the temperature of location A is different' (1 entity) or 'the temperature of A and B is different' (2 entities). If the entities are not specified, contextual information must be used.

A.3.1.4 OR2: Difference between quantities, adj. subpattern 2

Pattern:

```
<Entity1> [and <Entity2>] DIFF/ADJ <QType>
```

Instantiations:

(1) "If two cans having different depths are connected by a tube, volume will always flow toward the depth that is lower."
QType: depth
Entity1: (can)
Entity2: (can)

Characterization:

This pattern is a variation of adjective subpattern 1. The <QType> is a quantity type, e.g. temperature, depth etc. associated with the entities participating in the ordinal relation. DIFF/ADJ is an adjective expressing the inequality. In all instances of the corpus 'different' was used. One of the two entities is optional. We could have variations like 'the temperature is different' (no entities), 'the temperature of location A is different' (1 entity) or 'the temperature of A and B is different' (2 entities). If the entities are not specified, contextual information must be considered for determining the entities.

A.3.2 OR3, OR4: Comparison between quantities

These patterns use a real comparison between the quantities of two objects. They use a characteristic comparative construct ('greater than', 'hotter than' etc.). The actual comparative can be generic, i.e. quantity-neutral ('greater', 'less', 'more', etc.) or quantity-specific ('hotter', 'colder', 'denser', 'faster' etc.). Sentences following patterns with quantity-neutral comparatives must have the referenced quantity type explicitly mentioned, while in sentences with quantity-specific patterns the quantity type is determined by the comparative.

A.3.2.1 OR3: Quantity-neutral comparison, subpattern 1

Pattern:

```
<Entity1> <COMP/Qneutral> <QType> than <Entity2>
```

Instantiations:

(1) "Note that the narrow can has much less volume than the wide one."
QType: volume
Comparative: much less

Entity1: narrow can
Entity2: wide one

(2) "Even though the big pan of water has much more heat than the hot little stone, heat still flows from the stone into the water."
QType: heat
Comparative: much more
Entity1: big pan of water
Entity2: hot little stone

(3) "Hence, a body of water exhibits greater resistance to temperature change, called thermal stability, than does a land mass."
QType: resistance
Comparative: greater
Entity1: body of water
Entity2: land mass

(4) "Fast-moving molecules have more heat energy than slow-moving molecules."
QType: heat energy
Comparative: more
Entity1: fast moving molecules
Entity2: slow moving molecules

Characterization:
The <QType> is a quantity type, e.g. temperature, depth etc. associated with the entities participating in the ordinal relation. The entities are both required (because of the comparison). The comparative specifies a comparison between the two entities regarding the quantity. In this pattern, the comparative form of an adjective for a quantity-neutral ordinal relationship, e.g. 'more', 'less', 'greater', because the quantity type is explicitly stated.

A.3.2.2 OR3: Quantity-neutral comparison, subpattern 2

Pattern:

```
<Entity1> <Thing> VP <COMP/Qneutral>  than  <Entity2>
```

Instantiations:

(1) "Some substances conduct heat better and more rapidly than other substances."

Thing:	heat
Verb:	conduct
Comparative:	better
Entity1:	some substances
Entity2:	other substances

(2) "That is because some substances absorb heat energy more readily than other substances."

Thing:	heat energy
Verb:	absorb
Comparative:	better
Entity1:	some substances
Entity2:	other substances

Characterization:

This is a variation of the quantity-neutral comparison pattern, in which the quantity type is encoded in the verb. The quantity type in this pattern is *not* the <Thing>, even though in both sentences above the thing is actually a quantity associated with the two entities (heat, heat energy). The comparison however does not refer to heat but to some property associated with the verb - the conductivity (or flow rate) and the absorption rate of the substances (or the process?). The quantity type is encoded by the verb phrase in combination with the noun <Thing>.

A.3.2.3 OR3: Quantity-neutral comparison, subpattern 3

Pattern:

```
<Quantity1>  <COMP/Qneutral> than  <Quantity2>
```

Instantiations:

(1) "The depth of the water is higher than the depth of the hole, so the depth difference drives volume out through the hole."

QType:	depth
Comparative:	higher
Entity1:	water
Entity2:	hole

Characterization:
This is a variation of the quantity-neutral comparison pattern, in which the quantity type explicitly mentioned for each entity. The quantities have to be the same quantity type, otherwise the comparison wouldn't be valid. The comparative is a quantity-neutral adjective for this pattern.

A.3.2.4 OR4: Quantity-specific comparison

Pattern:

```
<Entity1> <COMP/Qspecific> than  <Entity2>
```

Instantiations:

(1) "If a small, hot stone is put into a big pan of warm water, heat will flow out of the stone and into the water, since the stone is hotter than the water."

Comparative: hotter
Entity1: stone
Entity2: water

(2) "Warm air is less dense than cold air so that the warm air rises and cooler, denser air sinks."

Comparative: less dense
Entity1: warm air
Entity2: cold air

(3) "Warm air near the surface of the Earth is heated by the Earth and becomes less dense than the cooler air above it."

Comparative: less dense
Entity1: warm air
Entity2: cooler air

(4) "Because the water is warmer than the ice, heat moves from the water to the ice."

Comparative: warmer
Entity1: water
Entity2: ice

(5) "So solid ice is less dense than liquid water."

Comparative: less dense
Entity1: solid ice
Entity2: liquid water

(6) "Gas molecules are already farther apart and moving faster than molecules in a solid or a liquid."

Comparative: moving faster
Entity1: gas molecules
Entity2: molecules in solid or liquid

Characterization:
The quantity is encoded in the adjective used in the comparison between the two entities. For example, in (1) the quantity referenced by 'hotter' is temperature. In sentence (2) it is 'density' (note that pattern 3 also applies to this sentence, 'hot' and 'cold' indicates an inequality of temperature between the two air masses). The adjective therefore cannot be quantity-neutral, i.e. it is quantity-specific (or quantity-dependent).

A.3.3 OR5: Adjective combination

This pattern does not include an explicit comparison (as in quantity-neutral or quantity-specific comparison patterns) or even a mentioning that two quantities are different. The comparison has to be constructed from a pair of adjectives. This includes both determining the quantity and finding the ordinal relationship between the entities associated with that quantity.

Pattern:

```
<Adj1/Qspecific> <Entity1>,   … <Adj2/Qspecific> <Entity2>
```

Instantiations:

(1) "Similarly, heat flows downhill from a higher temperature to a lower one."

Entity1: temperature
Adj1: higher
Entity2: one
Adj2: lower

(2) "Heat flows from hot things to cold things."

Entity1: things
Adj1: hot
Entity2: things
Adj2: cold

(3) "For instance, a silver spoon is hot when you've been stirring coffee because heat flows easily through the silver from the hot coffee to your fingers."

Entity1:	coffee
Adj1:	hot
Entity2:	fingers
Adj2:	-

(4) "As heat is conducted from the relatively warm ground to cooler overlying air, the air becomes warmer than the surrounding air."

Entity1:	ground
Adj1:	warm
Entity2:	air
Adj2:	cooler

(5) "They thought that heat was an invisible, weightless fluid capable of flowing from hotter objects to colder ones."

Entity1:	objects
Adj1:	hotter
Entity2:	ones
Adj2:	colder

(6) "If you have ever accidentally touched a hot pan, you have discovered for yourself (most likely in a painful way) that heat energy moves from a warmer object to a cooler object."

Entity1:	object
Adj1:	warmer
Entity2:	object
Adj2:	cooler

(7) "The movement of heat from a warmer object to a cooler one is called heat transfer."

Entity1:	object
Adj1:	warmer
Entity2:	one
Adj2:	cooler

(8) "Heat always moves from a warm substance to a cooler substance."

Entity1:	substance
Adj1:	warm
Entity2:	substance
Adj2:	cooler

.

(9) "As the more energetic molecules of the hot coffee collide with the less energetic atoms of the cooler spoon, some kinetic energy is transferred to the atoms of the spoon."

Entity1:	coffee
Adj1:	hot
Entity2:	spoon
Adj2:	cooler

(10) "The cooler air is then heated by the ground and the process is repeated."

Entity1:	air
Adj1:	cooler
Entity2:	ground
Adj2:	-

(11) "Warm air is less dense than cold air so that the warm air rises and cooler, denser air sinks."

Entity1:	air
Adj1:	warm
Entity2:	air
Adj2:	cold

Characterization:
Both entities are required, otherwise no comparison between two quantities could be constructed. The adjectives must be quantity-specific, and the quantity type will be derived from the adjectives. There is no explicit comparison. The ordinal relationship is constructed by a comparison of the adjectives belonging to the entities. There is a good chance that this pattern co-occurs with an Indirect Influence pattern. In fact, all of the sentences above also appear in a single pattern of the Indirect Influence analysis

A.4 Landmarks and limit points

The following is an analysis of the natural language patterns we found in our corpus material for describing limit points and landmark values. We first identified a number of key nouns, verbs, and prepositions that can refer to points and intervals. We then extracted the appropriate sentences and analyzed the NL patterns they encoded.

A.4.1 L1: Action at a point

Pattern:

```
<Point> <Condition>: <Action>
```

Instantiations:

(1) "At some point the depth gets so high that the water flows out of the can"

Point:	some point
Condition:	depth gets so high
Action:	water flow

(2) "Once this point is reached, the plate gets no hotter."

Point:	this point
Condition:	reached
Action:	plate gets no hotter

Characterization:

The <Point> is the landmark or limit point at which some <Action> happens given a <Condition>. This point can be referred to abstractly, as in the examples, or by a specific name, e.g. ‘boiling point’.

A.4.2 L2: Quantity at a point

Patterns:

```
<Quantity> <VP> AT <Point>
WHILE/DURING <Action>, <Quantity> <VP> AT <Point>
WHEN <Quantity> <VP> <Point>, <Action>
<Action> WHEN <Quantity> <VP> <Point>
```

Instantiations:

(1) "The temperature of the water remains at 150 degrees"
(2) "While the wax is melting, the temperature of the water stays at 150 degrees"
(3) "When the depth reaches the hole, water flows out"
(4) "The heater stops when the temperature reaches 72 degrees"
(5) "The temperature rises until it reaches 120 degrees"

Characterization:

This pattern is used for tying quantities to a particular point. The <Quantity> is described by a phrase containing an entity and the associated quantity type. In some cases, only the quantity type is mentioned and the entity has to be determined from contextual information.

A.4.3 L3: Conditional for points and intervals

Patterns:

```
<Conditional> <Quantity> <Point>, <Action>
<Conditional> <Quantity> <Interval>, <Action>
```

Instantiations:

(1) "While the temperature is under 32 degrees, the water remains in its solid form."
(2) "If its depth is above the hole, the fluid leaks out."
(3) "Within the boundary layer, heat is lost by conduction."
(4) "Outside the ..., ..."

Characterization:

The conditional is used to introduce a condition under which a <Quantity> at a specific <Point> or within some interval causes some <Action>. If the <Quantity> does not specify the associated entity and mentions only the quantity type, the entity has to be determined from contextual information.

A.4.4 L4: Labeling

Patterns:

```
<Value> is <Point>
<Point> is <Value>
```

Instantiations:

(1) "The temperature at with a substance changes from the liquid phase to the solid phase is called its freezing point"
(2) "32 degrees Fahrenheit is the freezing point of water"
(3) "The freezing point of water is 0 degrees Celsius"

Characterization:

This pattern is used for assigning a value of a particular point. The point can generally be treated as a landmark, and often even as a limit point.

Appendix B
Rewrite Material

This appendix lists the material used for the analysis of rewriting descriptions of physical phenomena from various sources in QRG-CE, as discussed in chapter 7. The following paragraphs show the original source text (**bold font**) as well as the rewritten version (indented, normal font).

B.1 Example 1

Source: (Buckley, 1979), ch.2, p.10-11

First, think of two different-sized cans that have the same depth of water in them.

The size of can C1 is different from the size of can C2.

The depth of water in can C1 is equal to the depth of water in can C2.

Though the depth of water in both cans is the same, the bigger can holds more volume.

The volume of can C1 is greater than the volume of can C2.

Similarly, if two pans of dirt are heated in an oven for several hours, both will have the same temperature.

Pan P1 contains dirt.

Pan P2 contains dirt.

Pan P1 is heated in an oven.

Pan P2 is heated in an oven.

The temperature of pan P1 is equal to the temperature of pan P2.

But even though the temperature of both pans is the same, the bigger pan holds more heat.

The heat of pan P1 is greater than the heat of pan P2.

Just as the depth can be the same while the volume is different, so can the temperature be the same while the heat is different.

The depth in can C1 can be the same as the depth in can C2.

The volume in can C1 can be different from the volume in can C2.

The temperature of pan P1 can be equal to the temperature of pan P2.

The heat of pan P1 can be different from the heat of pan P2.

Second, let's consider two objects in which the heat is the same yet the temperatures differ.

The heat of object O1 is equal to the heat of object O2.

The temperature of object O1 is different from the temperature of object O2.

Two warm bricks could contain the same amount of heat as one hot brick; though the hot brick naturally has a higher temperature than the warm ones.

Brick B1 is warm.
Brick B2 is warm.
Brick B3 is hot.
The heat of brick B1 and brick B2 is equal to the heat of brick B3.
The temperature of the hot brick B3 is greater than the temperature of the warm brick B1.
The temperature of the hot brick B3 is greater than the temperature of the warm brick B2.

In the same way, two cans can hold the same volume but have different depths.

The volume of can C1 is equal to the volume of can C2.
The depth of can C1 is different from the depth of can C2.

B.2 Example 2

Source: (Moran & Morgan, 1994), ch.3, p.59

Temperature is one of the most important and common variables used to describe the state of the atmosphere.

Temperature is a variable for describing the state of the atmosphere.

It is a usual component of weather reports and forecasts.

Temperature is a component of weather-reports.
Temperature is a component of weather-forecasts.

From everyday experience, we know that air temperature varies with time: from one season to another, between day and night, and even from one hour to the next.

The temperature of the air varies over time.

Air temperature also varies from one place to another: highlands and higher latitudes are usually colder than lowlands and lower latitudes.

The temperature of the air varies with places.

We also know that temperature and heat are closely related concepts.

Temperature is related to heat.

When we heat a pan of soup on the stove, the temperature of the soup rises.

The temperature of soup in a pan rises, because we heat the soup.

When we drop an ice cube into a beverage, the temperature of the beverage falls.

The temperature of a beverage falls, because we drop an ice-cube in the beverage.

B.3 Example 3

Source: (Moran & Morgan, 1994), ch.6, p.116

Water evaporates from the surface of seas, lakes, and rivers as well as from soil and the damp surfaces of plant leaves and stems.

Water evaporates from the surface of seas.
Water evaporates from the surface of lakes.
Water evaporates from the surface of rivers.
Water evaporates from the damp surfaces of leaves.
Water evaporates from the damp surfaces of stems.

Evaporation of ocean water is the principal source of atmospheric water vapor.

Evaporation of water from the ocean is the [primary] source of vapor in the atmosphere.
Vapor is the gaseous form of water.

Transpiration is the process by which water absorbed by plant roots eventually escapes as vapor through tiny pores on the surface of green leaves.

Plants absorb water through the roots.
Roots are a part of plants.
Green leaves transpire water [as vapor] through small pores.

On land, transpiration is considerable and is often more important than direct evaporation from the surfaces of lakes, streams, and the soil.

The amount of water from transpiration [on land] is greater than amount of water from evaporation.

For example, a single hectare (2.5 acres) of corn typically transpires 34,000 liters (L) (8800 gal) of water per day.

One hectare of corn transpires 34000 liters of water [daily].

Measurements of direct evaporation and transpiration are usually combined as evapotranspiration.

Evapotranspiration is a measurement [of evaporation and transpiration].

B.4 Example 4

Source: (Maton et al., 1994), ch.2, p.47

As sunlight strikes the collector, heat is absorbed.

Sunlight hits the collector.
Heat is absorbed.

The heat absorbed by the collector is transferred to the water.

Heat is absorbed by the collector.
The heat is transferred to the water.

The heated water flows through a tube into a storage tank.

The warm water flows through a tube into a tank.

Here the heat from the water in the tube is transferred to the water in the tank by a heat exchanger in the tank.

The heat is transferred from [the water in] the tube to [the water in] the tank by a heat-exchanger [in the tank].

The hot water circulates through pipes to heat the building or to heat air blown into the building.

The hot water circulates through pipes.

The water heats the building, or the water heats the air in the building.

In the meantime, a pump returns the cool water to the collector to be reheated by the sun.

A pump returns the cool water to the collector.

The sun reheats the cool water.

On cloudy days, when the solar collector cannot absorb enough solar energy to produce hot water and the storage system has cooled, a backup heating system is used.

A collector cannot absorb enough solar-energy on cloudy days.

A collector cannot produce enough hot water on cloudy days.

The tank cools on cloudy days.

A heater generates heat on cloudy days.

B.5 Example 5

Source: (Williams, 1992), ch.2, p.16

A heat engine depends on hot-cold contrasts to produce power.

A heat-engine produces power by a difference in temperature.

Your car is a good example.

Your car is a heat-engine.

Power comes from the heat created by burning a mixture of gasoline and air in a cylinder.

A cylinder contains gasoline.

The cylinder contains air.

The gasoline mixes with the air.

The mixture burns in a cylinder.

The burning of the mixture produces heat.

The heat generates power.

As the engine runs, cylinders alternate between hot and cooler.

The cylinders alternate between temperatures, while the engine runs

Power is produced.

The difference in temperature produces power.

B.6 Example 6

Source: (Lehr et al., 1987), p.8

The glass of a greenhouse lets the short solar rays pass through.
Short solar-rays penetrate the glass of a greenhouse
These are absorbed by objects inside and are re-radiated as long heat rays.
Objects in the greenhouse absorb the solar-rays.
The objects radiate long heat-rays.
But these long heat rays cannot get through the glass.
The long heat-rays cannot penetrate the glass.
The heat rays are continually re-absorbed and re-radiated inside.
The objects in the greenhouse absorb the heat-rays.
The objects radiate the heat-rays inside the greenhouse.
This helps keep the greenhouse warm on cold days.
The greenhouse is warm on cold days.
Some heat is lost by conduction through the glass.
The greenhouse loses heat through the glass by conduction.

B.7 Example 7

Source: (Lehr et al., 1987), p.13

Heat evaporates millions of tons of water into the air daily.
Heat evaporates millions of tons of water into the air daily.
Lakes, streams, and oceans send up a steady stream of water vapor.
Vapor is the gaseous form of water.
Lakes emit vapor to the air.
Streams emit vapor to the air.
Oceans emit vapor to the air.
An amazing amount of water transpires from the leaves of green plants.
Much water transpires from the leaves of green plants.
A single apple tree may move 1,800 gallons of water into the air in a six-month growing season.
A tree can emit 1800 gallons of water into the air [in six months].
As moist warm air rises, it slowly cools.
As the altitude of the air increases, the temperature of the air decreases.
Finally it cools so much that its relative humidity reaches 100 per cent.

The air cools, and the relative-humidity of the air reaches 100 percent.

Clouds form and, under certain conditions, rain or snow comes down.

Clouds form, because the temperature of the air reaches 100 percent.

Rain can fall, because clouds form.

Snow can fall, because clouds form.

This eternal process of evaporation, condensation, and precipitation is called the water cycle.

Evaporation is a part of the water-cycle.

Condensation is part of the water-cycle.

Precipitation is part of the water cycle.

B.8 Example 8

Source: (Gritzen, 1980), ch.1, p.6

As another example of the difference between heat and internal energy, consider two equal lengths of piping made of identical materials and containing steam at the same pressure and temperature.

The length of pipe p1 is equal to the length of pipe p2.

The material of pipe p1 is the same as the material of pipe p2.

Pipe p1 contains steam.

Pipe p2 contains steam.

The pressure of the steam in pipe p1 is equal to the pressure of the steam in pipe p2.

The temperature of the steam in pipe p1 is equal to the temperature of the steam in pipe p2.

One pipe is well-insulated; one is not.

Pipe p1 is insulated.

Pipe p2 is uninsulated.

From everyday experience, we expect more heat to flow from the uninsulated section of pipe than from the insulated section.

The flow of heat from pipe p1 is greater than the flow of heat from pipe p2.

When the two pipes are first filled with steam, the steam in one pipe contains exactly as much internal energy as the steam in the other.

The internal-energy of the steam in pipe p1 is equal to the internal-energy of the steam in pipe p2.

We know this is true because the two pipes contain equal volumes of steam at equal pressures and temperatures.

The volume of steam in pipe p1 is equal to the volume of steam in pipe p2.

B.9 Example 9

Source: (Gritzen, 1980), ch.1, p.7

A person sitting near a hot stove is warmed by thermal radiation from the stove, even though the air in between remains relatively cold.

A hot stove radiates heat.

The stove warms a person by thermal-radiation.

The air [between the person and the stove] is cold.

Thermal radiation from the sun warms the earth without warming the space through which it passes.

The sun warms the earth by thermal-radiation.

The sun does not warm the space [between the sun and the earth].

Thermal radiation passes through any transparent substance - air, glass, ice - without warming it to any extent because transparent materials are very poor absorbers of radiant energy.

Thermal-radiation passes through transparent substances.

Thermal-radiation does not warm transparent substances, because the substances do not absorb radiant-energy well.

B.10 Example 10

Source: (Schmidt, Henderson, & Wolgemuth, 1993), ch.1, p.9

As an illustration of this phenomena recall what happens when a can of soda is removed from the refrigerator and placed on a table.

A can is placed on a table.

The can contains cold soda.

The temperature of the soda starts to increase because of the flow of energy to it from the warmer air surrounding the can.

The temperature of the soda increases, because energy flows from the air to the soda.

Around computers it is difficult to find the correct unit of time to measure progress. Some cathedrals took a century to complete. Can you imagine the grandeur and scope of a program that would take as long?

-- Alan Perlis, Epigrams in Programming, ACM SIGPLAN, Sept. 1982